

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-296154

(43)Date of publication of application : 29.10.1999

(51)Int.Cl.

G09G 5/00

G06F 13/18

(21)Application number : 10-104342

(71)Applicant : HITACHI LTD

(22)Date of filing : 15.04.1998

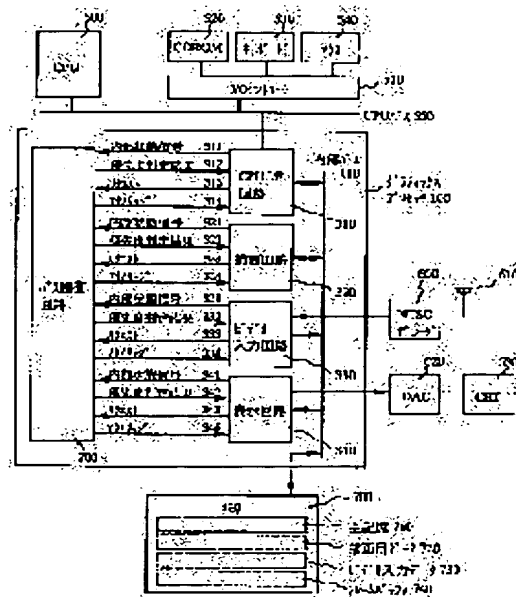
(72)Inventor : SHIMOMURA TETSUYA
MATSUO SHIGERU
KOGA KAZUYOSHI
KATSURA AKIHIRO
NAKATSUKA YASUHIRO
YAMAGISHI KAZUSHIGE

(54) DEVICE AND METHOD FOR GRAPHIC PROCESSING

(57)Abstract:

PROBLEM TO BE SOLVED: To conduct an arbitration so that plural circuits, which are required to complete their processes within a normal time, make necessary and sufficient accesses to a memory.

SOLUTION: A bus controlling circuit 200 compares the degree of urgency for the access to a memory 700 of each circuit, dynamically determines the priority of an internal bus 110 using internal state signals 311, 321, 331 and 341 outputted from a CPU I/F circuit 310, a plotting circuit 320, a video input circuit 330 and a display circuit 340 and conducts bus arbitration of the bus 110 using the priority. Thus, even though there exist plural circuits, which must complete their processes within a normal time, a necessary and sufficient memory access is conducted to complete the respective processes within the normal time.



LEGAL STATUS

[Date of request for examination] 29.09.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3497988

[Date of registration] 28.11.2003

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(51) Int.Cl. ⁶	識別記号	F I
G 0 9 G 5/00	5 5 5	G 0 9 G 5/00 5 5 5 P
G 0 6 F 13/18	5 1 0	G 0 6 F 13/18 5 1 0 A

審査請求 未請求 請求項の数23 O L (全 22 頁)

(21) 出願番号 特願平10-104342

(22) 出願日 平成10年(1998)4月15日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 下村 哲也

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(72) 発明者 松尾 茂

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(72) 発明者 古賀 和義

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(74) 代理人 弁理士 小川 勝男

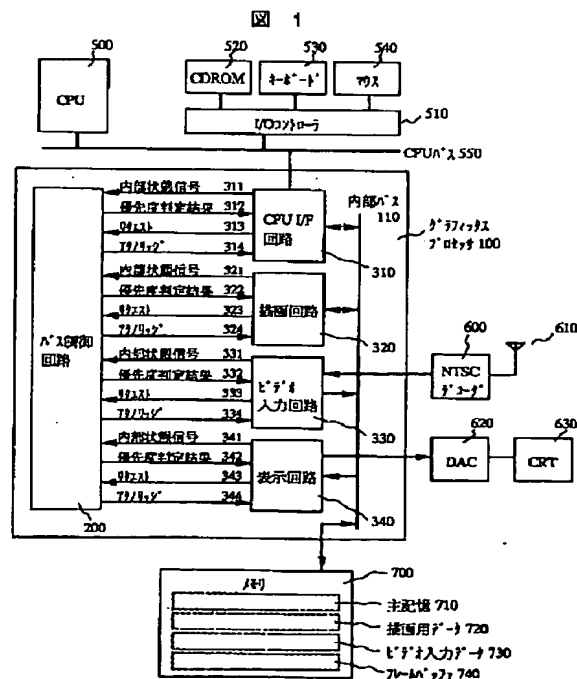
最終頁に続く

(54) 【発明の名称】 図形処理装置及び図形処理方法

(57) 【要約】

【課題】 ある規定時間内に必ず処理を終了する必要がある複数の回路がメモリへのアクセスを必要かつ十分に行うための調停を行うこと。

【解決手段】 CPU・I/F回路310、描画回路320、ビデオ入力回路330、表示回路340から出力される内部状態信号311、321、331、341を使って、バス制御回路200で各回路のメモリ700へのアクセスの緊急度を比較しながら内部バス110の優先度を動的に決定し、その優先度を用いて内部バス110のバス調停を行うことによって、規定時間内に必ず処理を終了しなければならない回路が複数ある場合でも、それぞれの規定時間内にそれぞれの処理を終了するのに必要な分だけのメモリアccessを行えるようにすることができる。



【特許請求の範囲】

【請求項1】 演算処理を実行するCPUと、
前記CPUの演算結果と表示データとを記憶するメモリと、
前記メモリにアクセスし、前記メモリに記憶された表示データの表示を制御する表示制御回路と、
表示データを生成するために前記メモリにアクセスして規定時間内に処理を行うための処理回路と、
前記表示制御回路及び前記処理回路から前記メモリへのアクセスの調停を行うバス制御回路とを有する図形処理装置であって、
前記表示制御回路、前記処理回路はそれぞれ内部の状態を内部状態信号として出力する手段を有し、前記バス制御回路は前記表示制御回路と前記処理回路から出力される前記内部状態信号に基づいて前記メモリにアクセスする際の優先度を判定することを特徴とする図形処理装置。

【請求項2】 請求項1の図形処理装置において、
前記CPUは、前記CPUに内蔵または外付けの周辺装置からの割込要求を前記バス制御回路に出力し、前記バス制御回路は前記割込要求を使って前記メモリにアクセスする際の優先度を判定することを特徴とする図形処理装置。

【請求項3】 請求項1または2の図形処理装置において、
前記CPUは前記割込要求を専用ピンを使って出力することを特徴とする図形処理装置。

【請求項4】 請求項1又は2の図形処理装置において、
前記CPUは前記割込要求をバスアクセスのプロトコルとして出力することを特徴とする図形処理装置。

【請求項5】 請求項1, 2, 3又は4の図形処理装置において、
前記表示データは1画素16ビットで構成された第1の表示データと、1画素8ビットで構成された第2の表示データであって、前記表示制御回路は前記第1の表示データの上に前記第2の表示データを任意の形状で重ね合わせて表示することを特徴とする図形処理装置。

【請求項6】 請求項1, 2, 3, 4又は5の図形処理装置において、
前記表示制御回路は前記表示データの一部をリニアアドレスで管理することを特徴とする図形処理装置。

【請求項7】 請求項1, 2, 3, 4, 5又は6の図形処理装置において、
前記表示制御回路は、前記表示データの一部を輝度信号及び色差信号によるデータフォーマットで管理することを特徴とする図形処理装置。

【請求項8】 請求項1, 2, 3, 4, 5, 6又は7の図形処理装置において、
前記処理回路は外部からビデオデータを入力し、入力した前記ビデオデータを輝度信号及び色差信号によるデー

タフォーマットまたはRGB形式のデータフォーマットの形式に変換して前記メモリに書き込むビデオ入力回路であることを特徴とする図形処理装置。

【請求項9】 請求項8の図形処理装置において、
前記ビデオ入力回路は、前記ビデオデータを前記メモリに書き込む際に前記ビデオデータの間引き処理及び補完処理を行うことを特徴とする図形処理装置。

【請求項10】 請求項1, 2, 3, 4, 5, 6, 7, 8又は9の図形処理装置において、
前記CPU, 前記メモリ, 前記表示制御回路, 前記処理回路は同一の半導体基板上に形成されていることを特徴とする図形処理装置。

【請求項11】 演算処理を実行するCPUと、
前記CPUの演算結果を記憶する第1のメモリと、
前記CPUの演算結果及び表示データとを記憶する第2のメモリと、
前記第2のメモリにアクセスし、前記第2のメモリに記憶された表示データの表示を制御する表示制御回路と、
表示データを生成するために前記第2のメモリにアクセスして規定時間内に処理を行うための処理回路と、
前記表示制御回路及び前記処理回路から前記第2のメモリへのアクセスの調停を行うバス制御回路とを有する図形処理装置であって、
前記表示制御回路、前記処理回路はそれぞれ内部の状態を内部状態信号として出力する手段を有し、前記バス制御回路は前記表示制御回路と前記処理回路から出力される前記内部状態信号に基づいて前記第2のメモリにアクセスする際の優先度を判定することを特徴とする図形処理装置。

【請求項12】 請求項11の図形処理装置において、
前記CPUは、前記CPUに内蔵または外付けの周辺装置からの割込要求を前記バス制御回路に出力し、前記バス制御回路は前記割込要求を使って前記第2のメモリにアクセスする際の優先度を判定することを特徴とする図形処理装置。

【請求項13】 請求項11または12の図形処理装置において、
前記CPUは前記割込要求を専用ピンを使って出力することを特徴とする図形処理装置。

【請求項14】 請求項11又は12の図形処理装置において、
前記CPUは前記割込要求をバスアクセスのプロトコルとして出力することを特徴とする図形処理装置。

【請求項15】 請求項11, 12, 13又は14の図形処理装置において、
前記表示データは1画素16ビットで構成された第1の表示データと、1画素8ビットで構成された第2の表示データであって、前記表示制御回路は前記第1の表示データの上に前記第2の表示データを任意の形状で重ね合わせて表示することを特徴とする図形処理装置。

【請求項 16】請求項 11、12、13、14 又は 15 の図形処理装置において、前記表示制御回路は前記表示データの一部をリニアアドレスで管理することを特徴とする図形処理装置。

【請求項 17】請求項 11、12、13、14、15 又は 16 の図形処理装置において、前記表示制御回路は、前記表示データの一部を輝度信号及び色差信号によるデータフォーマットで管理することを特徴とする図形処理装置。

【請求項 18】請求項 11、12、13、14、15、16 又は 7 の図形処理装置において、前記処理回路は外部からビデオデータを入力し、入力した前記ビデオデータを輝度信号及び色差信号によるデータフォーマットまたは RGB 形式のデータフォーマットの形式に変換して前記メモリに書き込むビデオ入力回路であることを特徴とする図形処理装置。

【請求項 19】請求項 18 の図形処理装置において、前記ビデオ入力回路は、前記ビデオデータを前記メモリに書き込む際に前記ビデオデータの間引き処理及び補完処理を行うことを特徴とする図形処理装置。

【請求項 20】請求項 11、12、13、14、15、16、17、18 又は 19 の図形処理装置において、前記 CPU、前記第 2 のメモリ、前記表示制御回路、前記処理回路は同一の半導体基板上に形成されていることを特徴とする図形処理装置。

【請求項 21】演算処理を実行する CPU と、前記 CPU の演算結果と表示データとを記憶するメモリと、前記 CPU と接続された第 1 の信号線、前記メモリと接続された第 2 の信号線、外部からの信号を取り込むための第 3 の信号線及び少なくとも前記 CPU を動作させるためのバッテリーと接続された第 4 の信号線と接続され、前記表示データを生成する描画部を有し、前記バッテリーの残量に応じて前記 CPU の演算結果、前記表示データ、前記外部信号の前記メモリへの書き込み又は読み出しの優先度を決定し、この優先度に基づいて前記メモリへのアクセスを行うグラフィックプロセッサとを有することを特徴とする図形処理装置。

【請求項 22】演算処理を実行する CPU、前記 CPU の演算結果と表示データとを記憶するメモリと接続されたグラフィックプロセッサが前記 CPU の演算結果又は生成した表示データの前記メモリへの書き込み及び読み出しを行う図形処理方法において、前記グラフィックプロセッサは、内部に蓄えられている前記メモリへ書き込むべき前記 CPU の演算結果及び前記生成した表示データの量及び前記読み出した表示データの量に基づいて前記メモリへのアクセスの優先度を決定することを特徴とする図形処理方法。

【請求項 23】少なくとも CPU からの命令によって生成された表示データ、外部から取り込んだビデオ又は音

声信号をメモリに書き込み、表示するために前記メモリに書き込まれた表示データ又はビデオデータ又は音声信号を前記メモリから読み出す図形処理方法において、CPU を動作させるためのバッテリーの残量に基づいて前記生成された表示データ又は外部から取り込んだビデオ信号又は音声信号の前記メモリへの書き込み又は読み出しの優先度を変更することを特徴とする図形処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、文字や図形データを発生、表示する図形処理装置に関するもので、特に表示用画素データを記憶するためのフレームバッファを主記憶装置内に一体化する、統合化メモリ方式で構成される図形処理装置及び方法に関するものである。

【0002】

【従来の技術】従来の技術による統合化メモリ方式で構成される図形処理装置の一例として、例えば特開平 4-1084192 号公報に記載された技術がある。この方式は、表示回路が他の回路に優先してメモリアクセスを行いたい場合に、そのことをメモリ管理ユニットに通知する優先制御信号を設け、メモリ管理ユニットは優先制御信号が入力されると表示回路のメモリアクセスの優先度を上げる制御を行うことで、メモリ統合化によりメモリアクセスの競合が増えても CRT などへの画面表示を途切れさせることなく制御する方式である。

【0003】

【発明が解決しようとする課題】上記発明は、ある規定時間内に必ず処理を終了する必要のある回路が、優先的にメモリアクセスを行うための手段として優先制御信号を設けたものである。

【0004】しかしながら上記発明では、ある規定時間内に必ず処理を終了する必要のある回路が複数存在する構成については考慮されていない。例えば、ある規定時間内に必ず処理を終了する必要のある回路として表示回路とビデオ入力回路を備える構成を考えると、表示回路は CRT への表示が途切れないようにメモリから表示データを読み出す必要があり、一方ビデオ入力回路についても、常に入力され続けるビデオデータを取りこぼさない用にビデオ入力データをメモリに書き込む必要がある。すなわち、表示回路は 1 画面を表示する時間内に 1 画面分全ての表示データをメモリから読み出す必要があり、ビデオ入力回路も同様にビデオ入力データが 1 画面分入力される時間内に 1 画面分全てのビデオ入力データをメモリに書き込む必要がある。このように表示回路とビデオ入力回路を備えた図形処理装置において上記発明を適用した場合、表示回路、ビデオ入力回路の両方に優先制御信号を設けることになるが、表示回路とビデオ入力回路が同時に優先制御信号をメモリ管理ユニットに出力すると、メモリ管理ユニットは結局いずれか一方の優先度の高い回路にメモリアクセスを行わせることになるの

で、もう一方の回路は優先制御信号を使っても、規定時間内に処理を終了するための十分なメモリアクセスを行えないことになり問題がある。

【0005】本発明の目的は、統合化メモリ方式を採用し、かつある規定時間内に必ず処理を終了する必要のある回路を複数備えた図形処理装置において、これらのある規定時間内に必ず処理を終了する必要のある複数の回路が、それぞれの規定時間内にそれぞれの処理を終了するのに必要な分だけメモリアクセスを行えるようにすることである。

【0006】

【課題を解決するための手段】上記目的は、演算処理を実行するCPUと、CPUの演算結果と表示データとを記憶するメモリと、メモリにアクセスし、メモリに記憶された表示データの表示を制御する表示制御回路と、表示データを生成するためにメモリにアクセスして規定時間内に処理を行うための処理回路と、表示制御回路及び処理回路からメモリへのアクセスの調停を行うバス制御回路とを有する図形処理装置であって、表示制御回路、処理回路はそれぞれ内部の状態を内部状態信号として出力する手段を有し、バス制御回路は表示制御回路と処理回路から出力される内部状態信号に基づいてメモリにアクセスする際の優先度を判定することによって達成することができる。

【0007】また上記目的は、演算処理を実行するCPUと、CPUの演算結果を記憶する第1のメモリと、CPUの演算結果及び表示データとを記憶する第2のメモリと、第2のメモリにアクセスし、第2のメモリに記憶された表示データの表示を制御する表示制御回路と、表示データを生成するために第2のメモリにアクセスして規定時間内に処理を行うための処理回路と、表示制御回路及び処理回路から第2のメモリへのアクセスの調停を行うバス制御回路とを有する図形処理装置であって、表示制御回路、処理回路はそれぞれ内部の状態を内部状態信号として出力する手段を有し、バス制御回路は表示制御回路と処理回路から出力される内部状態信号に基づいて第2のメモリにアクセスする際の優先度を判定することによって達成することができる。

【0008】

【発明の実施の形態】次に、本発明について図面を参照して詳細に説明する。

【0009】図1に本発明の第1の実施例の全体構成図を示す。図1において、100はグラフィックスプロセッサ、110は内部バス、200はバス制御回路、310はCPU I/F回路、320は描画回路、330はビデオ入力回路、340は表示回路、311、321、331、341は内部状態信号、312、322、332、342は優先度判定結果、313、323、333、343はリクエスト、314、324、334、344はアクノリッジ、500はCPU、510はI/O

コントローラ、520はCDROM、530はキーボード、540はマウス、550はCPUバス、600はNTSCデコーダ、610はアンテナ、620はDAC、630はCRT、700はメモリ、710は主記憶、720は描画用データ、730はビデオ入力データ、740はフレームバッファである。図1において、主記憶710はCPU500が動作するために必要なプログラム、データ、作業領域などを保持する領域であり、描画用データ720は描画回路320が動作するために必要なコマンド、データ、作業領域などを保持する領域であり、ビデオ入力データ730はビデオ入力回路330によって作成されたビデオ入力データを保持する領域であり、フレームバッファ740は表示回路340が読み出してCRT630に表示するための表示データを保持する領域である。グラフィックスプロセッサ100内の各回路について、以下説明を行う。CPU I/F回路310はCPU500からメモリ700へのアクセス要求を受けると、バス制御回路200に内部バス110のバス権を取得するためリクエスト313を出力する。CPU I/F回路310はバス制御回路200からアクノリッジ314が出力されるのを受けて、内部バス110を通してメモリ700にアクセスを行い、CPU500からのアクセス内容が読み出しアクセスであればメモリ700から読み出したデータをCPUバス550を通してCPU500に転送し、また書き込みアクセスであれば、メモリ700にそのデータの書き込みを行う。さらにCPU I/F回路310は、CPU500からメモリ700へのアクセス要求を受けてから現在までの経過時間を、内部状態信号311としてバス制御回路200に出力する。またCPU I/F回路310は、バス制御回路200から出力される優先度判定結果312の内容が、CPU I/F回路310の優先度が一番高いことを示している場合、直ちにリクエスト313を出力する。描画回路320は、描画用データ720から描画コマンドを読み出してグラフィック描画データを作成し、フレームバッファ740に書き込む。メモリ700にアクセスするために、リクエスト323とアクノリッジ324を使って内部バス110のバス権取得を行うのは、CPU I/F回路310と同様である。さらに描画回路320は、内部のバッファが空になってからの経過時間（読み出し用バッファの場合）、または一杯になってからの経過時間（書き込み用バッファの場合）を調べ、それらの中の最大値を内部状態信号321としてバス制御回路200に出力する。また描画回路320は、バス制御回路200から出力される優先度判定結果322の内容が、描画回路320の優先度が一番高いことを示している場合、直ちにリクエスト323を出力する。ビデオ入力回路330は、アンテナ610で受信後NTSCデコーダ600でデジタルデータに変換されたビデオデータを読み込む。ビデオ入力回路330は、ビデオデータのフォーマットとして、RGB形式ま

たは、輝度信号+色差信号形式（以下YC形式と呼ぶ）を選択することが可能である。また、入力されたビデオデータを間引いてデータサイズを小さくすることが可能である。これはピクチャー・イン・ピクチャーを行ってビデオ画面を子画面として元の画面サイズよりも小さいサイズで表示する場合に、ビデオデータの入力時の取り込み画面サイズを表示サイズに合わせることによってメモリ700を有効利用するために使用する。ビデオ入力回路330は、CRT630に表示する際の表示画面上でのビデオ画面の位置を示すレジスタを参照しながら、入力されたビデオデータをメモリ700に書き込む。ビデオ入力回路330は、入力されたビデオデータをメモリ700に書き込むため、リクエスト333とアクノリッジ334を使ってメモリ700へのアクセスを行う。さらにビデオ入力回路330は、内部のバッファがあとどれくらいで入力されるビデオデータで一杯になるかの予測時間を調べ、内部状態信号331としてバス制御回路200に出力する。またビデオ入力回路330は、バス制御回路200から出力される優先度判定結果332の内容が、ビデオ入力回路330の優先度が一番高いことを示している場合、直ちにリクエスト333を出力する。表示回路340は、メモリ700から表示データとビデオ入力データを読み出し、DAC620を通してCRT630に出力する。表示回路340は表示データを読み出す際に、CRT630での表示画面上でのビデオ画面の位置を示すレジスタを参照しながら、読み出した表示データまたはビデオ入力データをDAC620に出力する。表示回路340は、表示データとビデオ入力データをメモリ700から読み出すため、リクエスト343とアクノリッジ344を使ってメモリ700へのアクセスを行う。さらに表示回路340は、内部の各バッファがあとどれくらいで空になるかの予測時間を調べ、その最小値を内部状態信号341としてバス制御回路200に出力する。また表示回路340は、バス制御回路200から出力される優先度判定結果342の内容が、表示回路340の優先度が一番高いことを示している場合、直ちにリクエスト343を出力する。バス制御回路200は内部状態信号311、321、331、341によって、CPU I/F回路310、描画回路320、ビデオ入力回路330、表示回路340のメモリ700へのアクセスの緊急度を比較し、最も緊急度の高い回路が最も内部バス110のバス調停時の優先度が高くなるように優先度を決定し、優先度判定結果312、322、332、342として出力する。またバス制御回路200はこの優先度判定結果を使って、CPU I/F回路310、描画回路320、ビデオ入力回路330、表示回路340から出力されるリクエスト313、323、333、343の中から最も優先度の高い回路に対してアクノリッジを出力する。

【0010】上記のように、CPU I/F回路310、描画回路320、ビデオ入力回路330、表示回路

340から出力される内部状態信号311、321、331、341を使って、バス制御回路200で各回路のメモリ700へのアクセスの緊急度を比較しながら内部バス110の優先度を動的に決定し、その優先度を用いて内部バス110のバス調停を行うことによって、規定時間内に必ず処理を終了しなければならない回路が複数ある場合でも、それぞれの規定時間内にそれぞれの処理を終了するのに必要な分だけのメモリアクセスを行えるようにすることができる。

【0011】図2に図1におけるCPU I/F回路310の構成例を示す。図2において、2100はバッファ、2200はアクセス制御回路である。

【0012】図2において、CPU500からのアクセス要求があると、読み出しアクセス要求の場合はアドレスが、書き込みアクセス要求の場合はアドレスとデータがバッファ2100に書き込まれる。またCPU500からのアクセス要求はアクセス制御回路2200に入力され、アクセス制御回路2200ではアクセス要求が入力されてからの経過時間をカウントし、内部状態信号311として出力する。さらにアクセス制御回路2200は内部バス110を使用するための、リクエスト313の出力も行い、リクエスト313の出力中にアクノリッジ314が入力されると開始信号をバッファ2100に出力し、バッファ2100に保持されているアドレス（書き込みアクセスの場合にはデータも）を内部バス110に出力させる。アクセス制御回路2200はまた、優先度判定結果312を調べCPU I/F回路310の優先度が一番高いことが分かったと直ちにリクエスト313を出力し、アクノリッジ314が入力されると開始信号をバッファ2100に出力し、バッファ2100に保持されているアドレス（書き込みアクセスの場合にはデータも）を内部バス110に出力させる。

【0013】図3に図1における描画回路320の構成例を示す。図3において、3100は描画コア、3200はコマンドバッファ、3300はテキストチャバッファ、3400はライトバッファ、3500はアクセス制御回路である。

【0014】図3において、描画コア3100はコマンドバッファ3200から描画コマンドを読み出し、その描画コマンドを解析した結果テキストチャデータが必要であればテキストチャバッファ3300からテキストチャデータを読み出してグラフィックスデータを作成し、書き込みアドレスとともにライトバッファ3400に書き込む。コマンドバッファ3200とテキストチャバッファ3300は、保持しているコマンドまたはデータがなくなつてからの経過時間を内部状態信号としてアクセス制御回路3500に出力する。ライトバッファ3400は、描画コア3100からグラフィックスデータを書き込まれてからの経過時間を内部状態信号としてアクセス制御回路3500に出力する。アクセス制御回路3500で

10

20

30

40

50

は、これら3つのバッファから出力される内部状態信号から最も大きなものを選択して内部状態信号321として出力する。さらにアクセス制御回路3500は内部バス110を使用するためのリクエスト323の出力も行い、リクエスト323の出力中にアクノリッジ324が出力されると開始信号を描画コア3100に出力する。アクセス制御回路3500はまた、優先度判定結果322を調べ描画回路320の優先度が一番高いことが分かると直ちにリクエスト323を出力し、アクノリッジ324が入力されると開始信号を描画コア3100に出力し、内部バス110へのアクセスを実行させる。描画コア3100は、コマンドバッファ3200またはテキストバッファ3300のための読み出しアクセスならばアドレスを内部バス110に出力し、内部バス110を通して読み出したデータをコマンドバッファ320またはテキストバッファ3300のいずれかに書き込む。また描画コア3100は、ライトバッファ3400のための書き込みアクセスならば、ライトバッファ3400に保持してあるアドレスとデータを内部バス110に出力させる。

【0015】図4に図1におけるビデオ入力回路330の第1の構成例を示す。図4において、4100はフォーマット設定レジスタ、4200はフォーマット変換回路、4300はハードウェアウインドウレジスタ、4400は間引き回路、4500はブロックアドレス生成回路、4600はライトバッファ、4700はアクセス制御回路である。NTSCデコーダ600の出力フォーマットは、RGB形式である場合と、YC形式である場合と両方考えられるが、以下では少ないビット数でより多くの色数を表現できるYC形式で出力されるとして説明する。

【0016】図4において、フォーマット設定レジスタ4100は、ビデオ入力データをメモリ700に書き込む際のデータフォーマットを設定するレジスタである。フォーマット設定レジスタ4100には、RGB形式またはYC形式のいずれかを保持する。フォーマット変換回路4200は、NTSCデコーダ600から入力されるビデオデータを、16ビットのYC形式から16ビットのRGB形式に変換する回路である。NTSCデコーダ600からビデオデータが入力されると、フォーマット設定レジスタ4100の内容がYC形式であれば、入力されたビデオデータは16ビットYC形式のまま間引き回路4400に入力される。フォーマット設定レジスタ4100の内容がRGB形式であれば、入力されたビデオデータはフォーマット変換回路4200で16ビットのRGB形式に変換後、間引き回路4400に入力される。間引き回路4400では、入力されたビデオデータの横方向のデータの間引き処理と補間処理と、縦方向のデータの間引き処理と補間処理とを行って、ライトバッファ4600にビデオデータを出力する。ハードウイ

ンドウレジスタ4300は、CRT630での表示画面上でのビデオ画面の位置を保持するレジスタである。ブロックアドレス生成回路4500は、間引き回路4400から出力されるビデオデータに合わせて、ハードウェアウインドウレジスタ4300の内容を参照しながらメモリ700にビデオデータを書き込むためのブロックアドレスを生成し、ライトバッファ4600に出力する。ここでブロックアドレスとは、描画回路320のメモリアccessをなるべく高速にするため、例えば連続する512バイトのメモリ空間を、CRT630の画面上で横32画素×縦16画素×8ビットのブロック状領域になるようにアドレスを割り当てることである。このようなブロック状にアドレスを割り当てると、例えば縦の直線を描画した場合でも、1回のRASアドレス出力によってアクセスできるメモリライン内へ複数画素を書き込むことができるため高速化を図れる（連続するメモリ空間にCRT630の画面上の左上から右下へラスタスキャンする順番に画素を割り当てるリニアマッピングでは、縦の直線を描画されると、1回のRASアドレス出力によってアクセスできるメモリライン内へは1画素しか書き込むことができない）。ライトバッファ4600は、間引き回路4400とブロックアドレス生成回路4500から出力されるビデオデータとアドレスを保持する。またライトバッファ4600は、入力されるビデオデータが保持できるデータサイズと等しくなる予測時間を内部状態信号としてアクセス制御回路4700に出力する。さらにライトバッファ4600はアクセス制御回路4700から開始信号が入力されると、内部バス110に保持しているビデオデータとアドレスを出力する。アクセス制御回路4700では、ライトバッファから出力される内部状態信号を、内部状態信号331として出力する。さらにアクセス制御回路4700は、内部バス110を使用するためのリクエスト323の出力も行い、リクエスト323の出力中にアクノリッジ334が入力されると開始信号をライトバッファ4600に出力する。アクセス制御回路4700はまた、優先度判定結果322を調べビデオ入力回路330の優先度が高いことが分かると直ちにリクエスト333を出力し、アクノリッジ334が入力されると開始信号をライトバッファ4600に出力し、内部バス110へのアクセスを実行させる。

【0017】上記のように、ブロックアドレス生成回路4500を設け、ブロックアドレスを使って入力されたビデオデータをメモリ700に書き込むことによって、表示回路340でのアドレス生成のためのハードウェア増加を抑えながらビデオ入力を行うことができる。

【0018】図5に図1における表示回路340の第1の構成例を示す。図5において、5100はクロマキーカラーレジスタ、5200はハードウェアウインドウレジスタ、5300はフォーマット設定レジスタ、5400、5410はパレット変換回路、5420はフォーマ

10

20

30

40

50

ット変換回路、5500はアクセス制御回路、5600はメニューバッファ、5610は背景バッファ、5620はビデオバッファ、5700は表示制御回路、5800、5810は比較器、5820、5830はセクタである。図5の表示回路340は、複数画面の重ね合わせ表示を行うことができるもので、メニューバッファ5500は一番前に表示されるグラフィックスデータ（8ビット、インデックスカラー）を、背景バッファ5510は一番後ろに表示されるグラフィックスデータ（8ビット、インデックスカラー）を、ビデオバッファ5520はメニューバッファ5500に保持されるグラフィックス面と背景バッファ5510に保持されるグラフィックス面との間に表示されるビデオ入力データ（16ビット、YC形式またはRGB形式）を、それぞれ保持するバッファである。

【0019】図5において、クロマキーカラーレジスタ5100は、メニューバッファ5600に保持されているグラフィックスデータの中で、透明色として扱うパレット番号を保持するレジスタである。ハードウェアウインドウレジスタ5200は、CRT630での表示画面上でのビデオ画面の位置を保持するレジスタである。メニューバッファ5600に保持されているグラフィックスデータは、パレット変換回路5400で8ビットのインデックスカラーから16ビットのダイレクトカラーに変換される。同様に、背景バッファ5610に保持されているグラフィックスデータは、パレット変換回路5410で8ビットのインデックスカラーから16ビットのダイレクトカラーに変換される。ビデオバッファ5620に保持されているビデオ入力データは、フォーマット設定レジスタ5300の内容がYC形式であれば、フォーマット変換回路5420で16ビットのYC形式から16ビットのRGB形式に変換される。フォーマット変換レジスタ5300の内容がRGB形式であれば、そのまま出力される。比較器5810は、表示制御回路5700から出力される表示画素座標とハードウェアウインドウレジスタ5200の内容とを比較し、表示画素座標がビデオ画面内であればビデオ入力データを、そうでなければパレット変換回路5410から出力されたグラフィックスデータを、セクタ5380から出力させる。メニューバッファ5600に保持されているグラフィックスデータはまた比較器5800にも出力される。比較器5800は、メニューバッファ5600から出力されるグラフィックスデータとクロマキーカラーレジスタ5100の内容を比較し、メニューバッファ5600から出力されたグラフィックスデータが透明色であれば、セクタ5830から出力されるデータを、そうでなければパレット変換回路5400から出力されるデータを、セクタ5820からDAC620へ出力させる。メニューバッファ5600、背景バッファ5610、ビデオバッファ5620はまた、保持しているデータが空になる予測時間を内部

状態信号としてアクセス制御回路5500に出力する。アクセス制御回路5500では、これら3つのバッファから出力される内部状態信号からもっとも小さなものを選択して内部状態信号341として出力する。さらにアクセス制御回路5500は内部バス110を使用するためのリクエスト343の出力も行い、リクエスト343の出力中にアクノリッジ344が入力されると開始信号を表示制御回路5700に出力する。アクセス制御回路5500はまた、優先度判定結果342を調べ表示回路340の優先度が一番高いことが分かったと直ちにリクエスト343を出力し、アクノリッジ344が入力されると開始信号を表示制御回路5700に出力し、内部バス110へのアクセスを実行させる。表示制御回路5700は、CRT630でのラスタスキャンに同期して、CRT630での画面上での表示画素の座標を表示画素座標として出力する。また表示制御回路5700は、アクセス制御回路5500から開始信号を受けるとアドレスを内部バス110に出力し、内部バス110を通して読み出したデータをメニューバッファ5600または背景バッファ5610またはビデオバッファ5620に書き込む。

【0020】上記のように、クロマキーカラーレジスタ5100とハードウェアウインドウレジスタ5200を設け、クロマキーカラーレジスタ5100の内容とメニューバッファ5600の保持しているデータとを比較し、またハードウェアウインドウレジスタ5200の内容と表示画素座標を比較し、これらの比較結果に応じてメニューバッファ5600または背景バッファ5610またはビデオバッファ5620のいずれかからデータを読み出してCRT630に画面を表示することによって、8ビットのデータに16ビットのデータを重ねて表示し、さらにその上に任意の形状の8ビットデータを重ねて表示することができる。

【0021】図6に図1におけるバス制御回路200の第1の構成例を示す。図6(a)において、6100は優先度判定回路A、6200は優先度判定回路B、6300は優先度判定回路C、6400は調停回路である。

【0022】図6(a)において、優先度判定回路A6100には、内部状態信号331、341が入力される。優先度判定回路A6100は、これらの内部状態信号を比較し、ビデオ入力回路330と表示回路340とのいずれかの内部状態信号の示す値が小さい方を第1優先回路6101として出力し、残った方を第2優先回路6102として出力する。優先度判定回路B6200には、内部状態信号311、321が入力される。優先度判定回路B6200は、これらの内部状態信号を比較し、CPU I/F回路310と描画回路320とのいずれかの内部状態信号の示す値が大きい方を、第1優先回路6201として出力し、残った方を第2優先回路6202として出力する。優先度判定回路C6300は、第1優先回路6101が示す回路の優先度を1番、第2

優先度回路 6 1 0 2 が示す回路の優先度を 2 番、第 1 優先回路 6 2 0 1 が示す回路の優先度を 3 番、第 2 優先度回路 6 2 0 2 が示す回路の優先度を 4 番として優先度判定結果 3 1 2, 3 2 2, 3 3 2, 3 4 2 として出力する。優先度判定結果はまた、調停回路 6 4 0 0 にも出力される。調停回路 6 4 0 0 は、優先度判定結果に基づきリクエスト 3 1 3, 3 2 3, 3 3 3, 3 4 3 の優先度を判定し、内部バス 1 1 0 のバス権を与える回路にアクノリッジを出力する。

【0023】上記のように、優先度判定回路 C 6 3 0 0 において、優先度判定回路 A 6 1 0 0 で優先度を判定する回路を常に優先度判定回路 B 6 2 0 0 で優先度を判定する回路よりも優先度を高くすることによって、処理実行時に特に規定時間のない回路のメモリアクセス要求が原因で、ある規定時間内に必ず処理を終了する必要のある回路のメモリアクセスが待たされることを防ぐことができる。

【0024】図 7 に図 1 におけるバス制御回路 2 0 0 の第 2 の構成例を示す。図 7 (a) において、7 1 0 0 は優先度判定回路 A、7 1 1 0 はしきい値設定レジスタ、7 3 0 0 は優先度判定回路 C、7 5 0 0 は比較器である。

【0025】図 7 (a) において、優先度判定回路 A 7 1 0 0 は、内部状態信号 3 3 1, 3 4 1 を比較し、ビデオ入力回路 3 3 0 と表示回路 3 4 0 とのいずれかの内部状態信号の示す値が小さい方を第 1 優先回路 7 1 0 1 として出力し、残った方を第 2 優先回路 7 1 0 2 として出力する。また優先度判定回路 A 7 1 0 0 は、内部状態信号 3 3 1, 3 4 1 の示す値のうち、小さい方を緊急度として出力する。しきい値設定レジスタ 7 1 1 0 は、優先度判定回路 A 7 1 0 0 で優先度を判定する回路と、優先度判定回路 B 6 2 0 0 で優先度を判定する回路との優先度を切り替えるためのしきい値を保持するレジスタである。比較器 7 5 0 0 は、緊急度としきい値設定レジスタ 7 1 1 0 の値を比較し、緊急度の方が小さければ緊急信号として“1”を、そうでなければ“0”を出力する。優先度判定回路 C 7 3 0 0 は、図 7 (b) に示すように緊急信号が“1”であれば、第 1 優先回路 7 1 0 1 が示す回路の優先度を 1 番、第 2 優先回路 7 1 0 2 が示す回路の優先度を 2 番、第 1 優先回路 6 2 0 1 が示す回路の優先度を 3 番、第 2 優先回路 6 2 0 2 が示す回路の優先度を 4 番として、緊急信号が“0”であれば、第 1 優先回路 6 2 0 1 が示す回路の優先度を 1 番、第 2 優先回路 6 2 0 2 が示す回路の優先度を 2 番、第 1 優先回路 7 1 0 1 が示す回路の優先度を 3 番、第 2 優先回路 7 1 0 2 が示す回路の優先度を 4 番として、優先度判定結果 3 1 2, 3 2 2, 3 3 2, 3 4 2 を出力する。

【0026】上記のように、しきい値設定レジスタ 7 1 1 0 を設け、しきい値設定レジスタ 7 1 1 0 の保持内容と優先度判定回路 A 7 1 0 0 から出力される緊急度と比較し、その結果優先度判定回路 A 7 1 0 0 で優先度を判

定する回路がすぐにメモリアクセスできなくとも問題ないと判断できる場合には、優先度判定回路 B 6 2 0 0 で優先度を判定する回路の優先度を優先度判定回路 A 7 1 0 0 で優先度を判定する回路の優先度よりも高くすることによって、ある規定時間内に必ず処理を終了する必要のある回路の方に時間的な余裕がある場合に、処理時間に特に規定時間のない回路のメモリアクセスを優先して行わせることができるため、ある規定時間内に必ず処理を終了する必要のある回路のメモリアクセスを十分行いながら、図形処理装置全体の性能を向上できる。

【0027】図 8 に図 1 におけるビデオ入力回路 3 3 0 の第 2 の構成例を示す。図 8 において、8 5 0 0 はリニアアドレス生成回路である。

【0028】図 8 において、リニアアドレス生成回路 8 5 0 0 は、間引き回路 4 4 0 0 から出力されるビデオデータに合わせて、ハードウェアウインドウレジスタ 4 3 0 0 の内容を参照しながらメモリ 7 0 0 にビデオデータを書き込むためのリニアアドレスを生成し、ライトバッファ 4 6 0 0 に出力する。ライトバッファ 4 6 0 0 は、アクセス制御回路 4 7 0 0 から出力される開始信号を受けて、内部バス 1 1 0 にアドレスとデータを出力する。

【0029】上記のように、リニアアドレス生成回路 8 5 0 0 を設け、リニアアドレスを使って入力されたビデオデータをメモリ 7 0 0 に書き込むことによって、メモリ 7 0 0 を有効に使用することができる。これは例えば連続する 5 1 2 バイトのメモリ空間を、CRT 630 の画面上で横 1 6 画素×縦 1 6 画素×1 6 ビットのブロック状領域になるようにアドレスを割り当てた場合を考えると、入力されるビデオデータのサイズが横 1 6 画素×縦 1 6 画素のブロックの整数倍である横 3 2 0 画素×縦 2 4 0 画素の場合にはブロックアドレスでも無駄なくメモリ上にマッピングできるが、横 1 0 0 画素×8 0 画素の場合には横 1 6 画素×縦 1 6 画素のブロックの整数倍にならないため、ブロックアドレスではメモリ上にマッピングする際に無駄ができてしまう。一方、リニアマッピングでは、入力されるビデオデータのサイズがどのような値であっても無駄なくメモリ上にデータをマッピングできる。したがって、ブロックマッピングに比べてリニアマッピングの方がメモリを有効に使うことができる。

【0030】図 9 に図 1 における表示回路 3 4 0 の第 2 の構成例を示す。図 9 において、9 7 0 0 は表示制御回路である。

【0031】図 9 において、表示制御回路 9 7 0 0 は、CRT 630 でのラスタスキャンに同期して、CRT 630 での画面上での表示画素の座標を表示画素座標として出力する。また表示制御回路 5 7 0 0 は、アクセス制御回路 5 5 0 0 から開始信号を受けるとメニューバッファ 5 6 0 0 または背景バッファ 5 6 1 0 用のメモリアクセスを行う場合にはブロックアドレスを、ビデオバッファ 5 6 2 0 用のメモリアクセスを行う場合にはリニアアドレスを内部

バス 110 に出力し、内部バス 110 を通して読み出したデータをメニューバッファ 5600 または背景バッファ 5610 またはビデオバッファ 5620 に書き込む。

【0032】上記のように、表示制御回路 9700 でアドレス生成する際に、アクセスするデータの種類のによってブロックアドレスかリニアアドレスかを選択して生成することによって、メモリ 700 を有効に使用することができる。

【0033】図 10 に本発明の第 2 の実施例の全体構成図を示す。図 10 において、10100 はグラフィックスプロセッサ、10200 はバス制御回路、10500 は CPU、10510 はシステムタイマー、10520 はユーザタイマー、10530 は DMAC、10540 は割り込み制御回路、10550 は CPU コア、10560 はメモリ I/F 回路、10600 は I/O コントローラである。

【0034】図 10 において、割り込み制御回路 10540 は、システムタイマー 10510、ユーザタイマー 10520、DMAC 10530 からそれぞれ出力される内部割り込み要求と、I/O コントローラ 10600 から出力される外部割り込み要求とを受け付け、割り込み通知として出力する。バス制御回路 10200 は、内部状態信号 311、321、331、341 と、割り込み通知を使って、そのタイミングでの内部バス 110 を使用するための優先度を決定し、優先度判定結果 10312、10322、10332、10342 として出力する。

【0035】上記のように、内部状態信号 311、321、331、341 と割り込み通知を使ってバス制御回路 10200 が内部バス 110 の優先度を決定することによって、CPU 10500 の内部または外部の周辺回路からの割り込み要求によって CPU 10500 のメモリ 700 へのアクセスの優先度を上げることができる。これは例えば、システムタイマー 10510 が CPU コア 10550 の暴走を検出して割り込み要求を出力した場合に、バス制御回路 10200 が割り込み通知を受けて CPU I/F 回路 310 の優先度を一番高くすることができ、速やかに暴走した CPU 10500 の復帰処理を行うことができる。なお図 10 の構成例では、CPU 10500 からのグラフィックスプロセッサ 10100 への割り込み通知の出力を専用のピンを使用する構成になっているが、これを CPU 10500 から CPU バス 550 を介してグラフィックスプロセッサ 10100 をアクセスする際のアクセスプロトコルとすることも可能である。

【0036】図 11 に図 10 におけるバス制御回路 10200 の構成例を示す。図 11(a) において、11200 は優先度判定回路 B、11210 は優先割り込みレジスタ、11220 は比較器、11300 は優先度判定回路 C である。

【0037】図 11(a) において、優先割り込みレジスタ 11210 は、CPU 10500 から出力される割り込み通

知のなかで、CPU I/F 回路 310 の内部バス 110 の優先度を一番高く設定したい割り込み処理を保持するレジスタである。比較器 11220 は、優先割り込みレジスタ 11210 の内容と割り込み通知を比較し、等しければ CPU 優先信号として “1” を、そうでなければ “0” を出力する。優先度判定回路 B 11200 には、内部状態信号 311、321 が入力される。優先度判定回路 B 11200 は、これらの内部状態信号を比較し、CPU 優先信号が “0” の場合には CPU I/F 回路 310 と描画回路 320 とのいずれかの内部状態信号の示す値が大きい方を第 1 優先回路 11201 として出力し、残った方を第 2 優先回路 11202 として出力する。一方、CPU 優先信号が “1” の場合には、CPU I/F 回路 310 を第 1 優先回路 11201 として出力し、描画回路 320 を第 2 優先回路 11202 として出力する。優先度判定回路 C 11300 は、緊急信号と CPU 優先信号によって図 11(b) のように各回路の優先度を判定し、優先度判定結果 10312、10322、10332、10342 として出力する。優先度判定結果はまた、調停回路 6400 にも出力される。

【0038】図 12 に本発明の第 3 の実施例の全体構成図を示す。図 12 において、12100 はグラフィックスプロセッサ、12200 はバス制御回路、12350 はサウンド回路、12360 は通信回路、12500 は CPU、12640 は DAC、12650 はスピーカ、12660 はモデム、12670 は通信回線、12700 はメモリ、12800 はバッテリーである。

【0039】図 12 において、バス制御回路 12200 は、内部状態信号 311、321、331、341、12351、12361 とバッテリー 12800 から入力されるバッテリー残量を使って、そのタイミングでの内部バス 110 の優先度を決定し、優先度判定結果 12312、12322、12332、12342、12352、12362 として CPU 12500 を含めた各回路に出力される。優先度判定結果 12312、12322、12332、12342、12352、12362 はまた、メモリアccess方法に関する情報も含んでおり、各回路は、優先度判定結果 12312、12322、12332、12342、12352、12362 に含まれるメモリアccess方法に関する情報に基づいてメモリアccess方法をバッテリー残量に応じて変更する。

【0040】上記のように、バス制御回路 12200 において内部状態信号 311、321、331、341、12351、12361 とバッテリー 12800 から入力されるバッテリー残量を使って内部バス 110 の優先度を決定し、その結果にメモリアccess方法に関する情報を含めて優先度判定結果 12312、12322、12332、12342、12352、12362 として CPU 12500 も含めて出力することで、バッテリー 12800 の残量に応じて内部バス 110 の優先度を制御し、バッテリー

残量が少なくなると低消費電力化するように内部バス110のアクセスを制御することができる。なお図12の構成例では、CPU12500からのグラフィックスプロセッサ12100への割り込み通知の出力を専用のピンを使用する構成になっているが、これをCPU12500からCPUバス550を介してグラフィックスプロセッサ12100をアクセスする際のアクセスプロトコルとすることも可能である。

【0041】図13に図12におけるバス制御回路12200の第1の構成例を示す。図13において、13100は優先度判定回路A、13110はしきい値設定レジスタ、13300は優先度判定回路C、13400は調停回路、13500はバースト長テーブル、13600はシフト回路である。

【0042】図13において、優先度判定回路A13100は、内部状態信号331、341、12351、12361を受け、その値が小さい順に、優先度を高い方から割り当てて行き、第1優先回路13101から第4優先回路13104までを出力する。優先度判定回路A13100はまた、内部状態信号331、341、12351、12361が示す値の中から、一番小さい値を緊急度として出力する。優先度判定回路C13300は、緊急信号が

“1”であれば第1優先回路13101が示す回路の優先度を一番とし、以下第2優先回路13102、第3優先回路13103、第4優先回路13104、第1優先回路6201、第2優先回路6202の順で優先度を割り当て、緊急信号が“0”であれば第1優先回路6201が示す回路の優先度を一番とし、以下第2優先回路6202、第1優先回路13101、第2優先回路13102、第3優先回路13103、第4優先回路13104の順で優先度を割り当て、優先度判定結果として出力する。優先度判定結果はまた、調停回路13400にも出力される。バースト長テーブル13500は、バッテリー残量に応じた各回路のメモリアクセス時のバースト長を保持するものであり、バッテリー残量が少なくなるとメモリアクセスを低消費電力化するようにバースト長を長くするための情報が保持されている。バッテリー12800からバッテリー残量が入力されると、その値に応じて各回路のバースト長と最長アクセスサイクル数を選択して出力する。ここで出力されるバースト長は、優先度判定回路C13300が出力する優先度判定結果とあわせ、優先度判定結果12312、12322、12332、12342、12352、12362として出力される。またバースト長テーブル13500から出力される最長アクセスサイクル数は、シフト回路13600で2倍された後でしきい値設定レジスタ13110に入力され、しきい値設定レジスタ13110の保持している値を更新する。

【0043】上記のように、バースト長テーブル13500を設け、バッテリー残量によって各回路のバースト長

を選択して優先度判定結果12312、12322、12332、12342、12352、12362に含めて出力することによって、バッテリー残量が少なくなるとより低消費電力化してメモリアクセスを行わせることができる。またこのような制御を行っても、バッテリー残量に応じてしきい値設定レジスタ13110の内容を更新することで、規定時間内に必ず処理を終了しなければならない回路が複数ある場合でも、それぞれの規定時間内にそれぞれの処理を終了するのに必要な分だけのメモリアクセスを行えるようにすることができる。

【0044】図14にバースト長=8、16のときの読み出しアクセス用コマンド内訳を示す。図14は、データを16個読み出す例を示してある。

【0045】図14において、(a)バースト長=8のときには、1回のバーストアクセスでデータを8個読み出すため、データを16個読み出すためには2回バーストアクセスを行う必要がある。1回のバーストアクセスあたり、バンクアクティブ：1回、カラムアドレス8回、プリチャージ：1回必要なので、合計バンクアクティブ：2回、カラムアドレス16回、プリチャージ：2回必要である。一方、(b)バースト長=16のときには、1回のバーストアクセスでデータを16個読むことができるため、バーストアクセス1回で良いことになる。この場合、バンクアクティブ：1回、カラムアドレス16回、プリチャージ：1回で良い。したがって、バーストアクセス=16にすることで、バンクアクティブ：1回、プリチャージ：1回のコマンドを実行しなくても良いため、その分バスをドライブする回数が減り低消費電力化できる。またバンクアクティブやプリチャージのコマンドをメモリ側で実行する必要がなくなるため、メモリ自身としても低消費電力化できる。

【0046】図15に図12におけるバス制御回路12200の第2の構成例を示す。図15において、15500は動作モードテーブルである。

【0047】図15において、動作モードテーブル15500は、バッテリー残量に応じた各回路の動作モードを保持するものである。各回路の動作モードとして、CPU/F回路310用はCPU12500の動作クロック（例えばバッテリーのフル充電時のCPU12500が50MHzであれば、バッテリー残量が減るのにしたがって37.5MHz、25MHz、25MHzというように下がる）、描画回路320用は動作クロック、ビデオ入力回路330用は入力されるビデオ画面のフレームレート、表示回路340用はCRT630に表示する表示画面のフレームレート、サウンド回路12350用は再生するサウンドのサンプリング周波数、通信回路12360用はモデム12660の通信速度が保持されている。動作モードテーブル15500は、バッテリー12800から入力されるバッテリー残量によって、各回路の動作モードを選択し、優先度判定回路C13300から出力される優先度判定結果とあわせ

て優先度判定結果信号12312, 12322, 12332, 12342, 12352, 12362として出力される。

【0048】上記のように、動作モードテーブル15500を設け、バッテリー残量によって各回路の動作モードを選択し、優先度判定結果信号12312, 12322, 12332, 12342, 12352, 12362に含めて出力することによって、バッテリー残量が減るにつれてバッテリー消費が少なくなるように各回路の消費電力を徐々に下げようように制御することができる。このよう

な制御を行うことで、バッテリー残量がある値以下になったときに、特定の回路の機能が突然使えなくなるのではなく、徐々に機能を限定していくようにすることができるようになり、利用者の利便性を改善できる。

【0049】図16に本発明の第4の実施例の全体構成図を示す。図16において、16700, 16800はメモリである。

【0050】図16において、メモリ16800は主記憶16810を保持するメモリで、メモリ16700は描画用データ16720、ビデオ入力データ16730、フレームバッファ16740を保持するメモリである。このような構成においても、バス制御回路200が各回路の内部状態信号に基づいて内部バス110の優先度を判定してバス調停を行うことで、規定時間内に必ず処理を終了しなければならない回路が複数ある場合でも、それぞれの規定時間内にそれぞれの処理を終了するのに必要な分だけのメモリアクセスを行えるようにすることができる。

【0051】図17に本発明の第5の実施例の全体構成図を示す。図17において、17100はグラフィックスプロセッサ、17500はCPU、17510はI/Oコントローラ、17700はメモリである。

【0052】図17において、グラフィックスプロセッサ17100は、CPU17500、I/Oコントローラ17510、メモリ17700も含んで1チップ化した構成になっている。このように全て1チップ化すると、メモリバス17110の動作周波数の高速化、バス幅のワイド化などが可能になり、メモリ17700のスループットは大幅に改善される。しかしある短いタイミングで考えると、やはりビデオ入力回路330と表示回路340が同時にメモリバス17110のアクセス要求を出力し、その際どちらか一方がメモリアクセスできなくなる状況が発生し得る。しかしバス制御回路200が内部状態信号を使ってメモリバス17110の優先度を判定してバス調停を行うことで、規定時間内に必ず処理を終了しなければならない回路が複数ある場合でも、それぞれの規定時間内にそれぞれの処理を終了するのに必要な分だけのメモリアクセスを行えるようにすることができる。

【0053】図18に本発明の第6の実施例の全体構成図を示す。図18において、18100はグラフィックス

ロセッサ、18200はバス制御回路である。

【0054】図18において、バス制御回路18200は、各回路から入力される内部状態信号とバッテリー12800から入力されるバッテリー残量に基づき内部バス110の優先度を判定し、その結果にメモリアクセス方法に関する情報を含めて優先度判定結果として出力する。このような構成にすることで、バッテリー12800の残量に応じて内部バス110の優先度の制御し、バッテリー残量が少なくなると低消費電力化するように内部バス110のアクセスを制御することができる。なお図18の構成例では、CPU12500からのグラフィックスプロセッサ18100への割り込み通知の出力を専用のピンを使用する構成になっているが、これをCPU12500からCPUバス550を介してグラフィックスプロセッサ18100をアクセスする際のアクセスプロトコルとすることも可能である。

【0055】図19に本発明の第7の実施例の全体構成図を示す。図19において、19100はグラフィックスプロセッサ、19500はCPUである。

【0056】図19において、グラフィックスプロセッサ19100は、CPU19500、I/Oコントローラ17510、メモリ17700も含んで1チップ化した構成になっている。このように全て1チップ化すると、CPU、メモリ、グラフィックスプロセッサを別チップで構成したときに比べ、チップ間を結ぶ配線をドライブする必要がなくなるため、低消費電力化できる。しかしながら、バス制御回路18200が、各回路から入力される内部状態信号とバッテリー12800から入力されるバッテリー残量に基づき内部バス110の優先度を判定し、その結果にメモリアクセス方法に関する情報を含めて優先度判定結果として出力することによって、バッテリー12800の残量に応じて内部バス110の優先度を制御し、バッテリー残量が少なくなると低消費電力化するように内部バス110のアクセスを制御することができるため、単に1チップした場合に比べてより一層低消費電力化を図ることができる。なお図19の構成例では、CPU19500からのバス制御回路18200への割り込み通知の出力を専用の配線を使用する構成になっているが、これをCPU19500からCPUバス17550を介してCPU I/F回路310をアクセスする際のアクセスプロトコルとすることも可能である。

【0057】

【発明の効果】以上本発明によれば、ビデオ入力回路330、表示回路340を備えたグラフィックスプロセッサ100のように、規定時間内に必ず処理を終了しなければならない回路が複数ある場合でも、それぞれの規定時間内にそれぞれの処理を終了するのに必要な分だけのメモリアクセスを行えるようにすることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施例の全体構成図である。

【図2】CPU I/F回路310の構成例である。
 【図3】描画回路320の構成例である。
 【図4】ビデオ入力回路330の第1の構成例である。
 【図5】表示回路340の第1の構成例である。
 【図6】バス制御回路200の第1の構成例である。
 【図7】バス制御回路200の第2の構成例である。
 【図8】ビデオ入力回路330の第2の構成例である。
 【図9】表示回路340の第2の構成例である。
 【図10】本発明の第2の実施例の全体構成図である。
 【図11】第2の実施例におけるバス制御回路1020
 0の構成例である。
 【図12】本発明の第3の実施例の全体構成図である。
 【図13】第3の実施例におけるバス制御回路1220
 0の第1の構成例である。
 【図14】バースト長=8, 16のときの読み出しアクセ
 ス用コマンド内訳である。
 【図15】第3の実施例におけるバス制御回路1220
 0の第2の構成例である。
 【図16】本発明の第4の実施例の全体構成図である。
 【図17】本発明の第5の実施例の全体構成図である。
 【図18】本発明の第6の実施例の全体構成図である。
 【図19】本発明の第7の実施例の全体構成図である。
 【符号の説明】

100, 10100, 12100, 17100, 181
 00, 19100…グラフィックスプロセッサ、110
 …内部バス、200, 10200, 12200, 182
 00…バス制御回路、310…CPU I/F回路、3
 20…描画回路、330…ビデオ入力回路、340…表
 示回路、311, 321, 331, 341…内部状態信
 号、312, 322, 332, 342…優先度判定結
 果、313, 323, 333, 343…リクエスト、3
 14, 324, 334, 344…アクノリッジ、50
 0, 10500, 12500, 17500, 19500
 …CPU、510, 10600, 17510…I/Oコ

【図2】

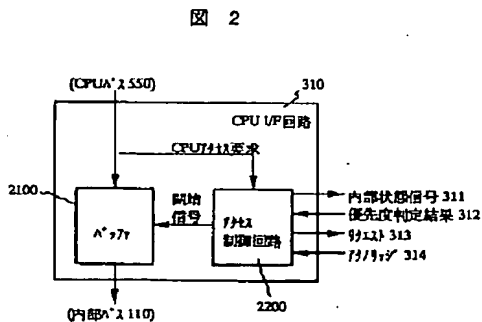


図 2

ントローラ、520…CDROM、530…キーボー
 ド、540…マウス、550…CPUバス、600…NT
 SCデコーダ、610…アンテナ、620, 12640…
 DAC、630…CRT、700, 12700, 167
 00, 16800, 17700…メモリ、710…主記
 憶、720…描画用データ、730…ビデオ入力デー
 タ、740…フレームバッファ、2100…バッファ、
 2200, 3500, 4700, 5500…アクセス制
 御回路、3100…描画コア、3200…コマンドバッ
 ファ、3300…テクスチャバッファ、3400, 46
 00…ライトバッファ、4100, 5300…フォーマット
 設定レジスタ、4200, 5420…フォーマット変換
 回路、4300, 5200…ハードウェアウインドウレ
 ジスタ、4400…間引き回路、4500…ブロックア
 ドレス生成回路、5100…クロマキーカラーレジス
 タ、5400, 5410…パレット変換回路、5600
 …メニューバッファ、5610…背景バッファ、5620…
 ビデオバッファ、5700, 9700…表示制御回路、
 5800, 5810, 7500, 11220…比較器、
 5820, 5830…セクタ、6100, 7100,
 13100…優先度判定回路A、6200, 11200
 …優先度判定回路B、6300, 7300, 1130
 0, 13300…優先度判定回路C、6400, 134
 00…調停回路、7110, 13110…しきい値設定
 レジスタ、8500…リニアアドレス生成回路、105
 10…システムタイマー、10520…ユーザタイマ
 ー、10530…DMAC、10540…割り込み制御回
 路、10550…CPUコア、10560…メモリ I/
 F回路、11210…優先割り込みレジスタ、1235
 0…サウンド回路、12360…通信回路、12650
 …スピーカ、12660…モデム、12670…通信
 回線、12800…バッテリー、13500…バースト長
 テーブル、13600…シフト回路、15500…動作
 モードテーブル。

【図3】

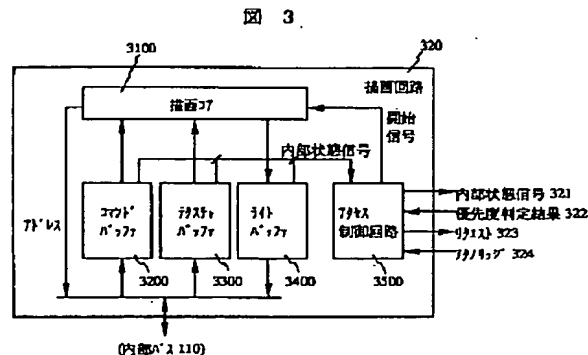
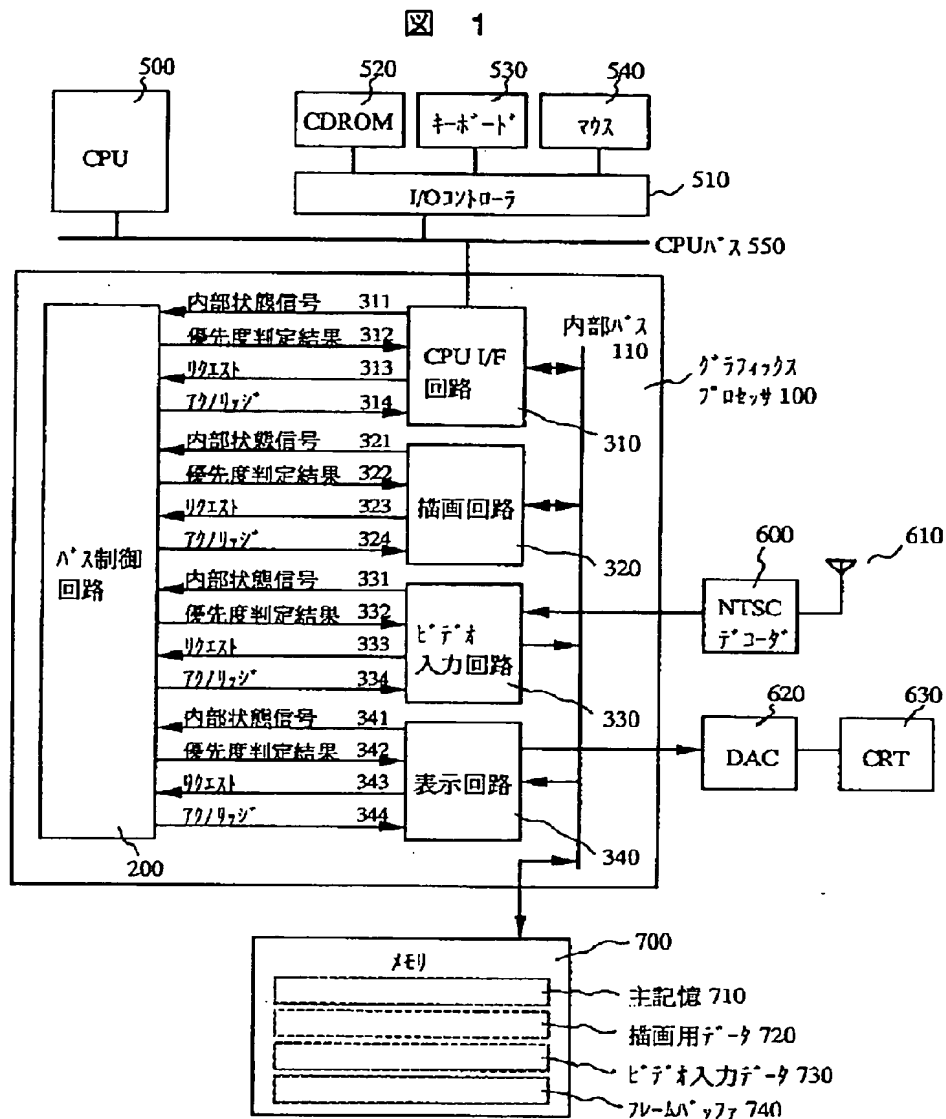


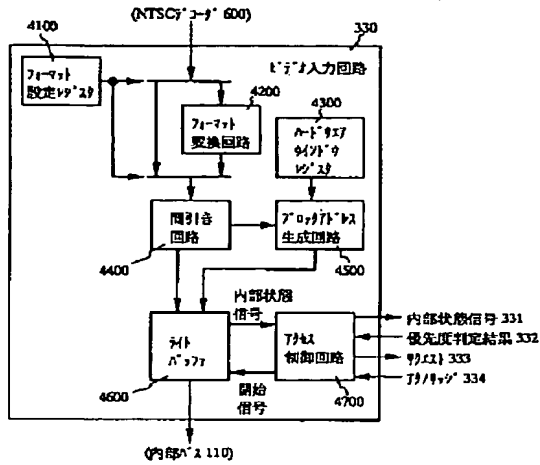
図 3

【図1】



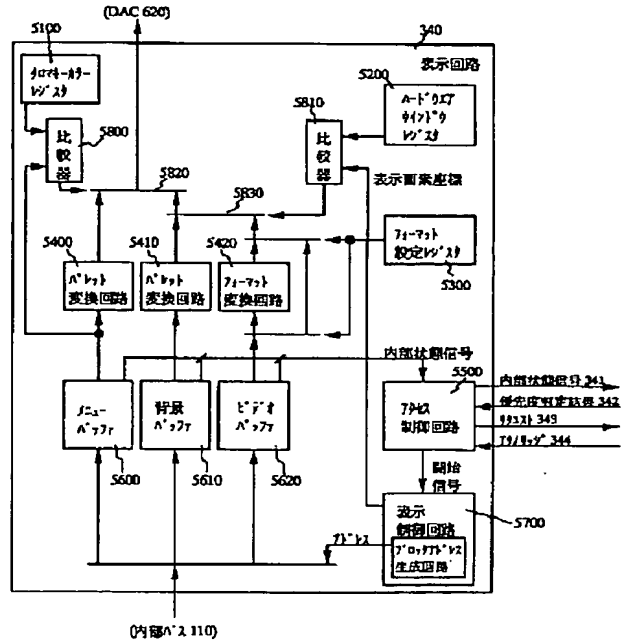
【図4】

圖 4



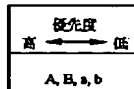
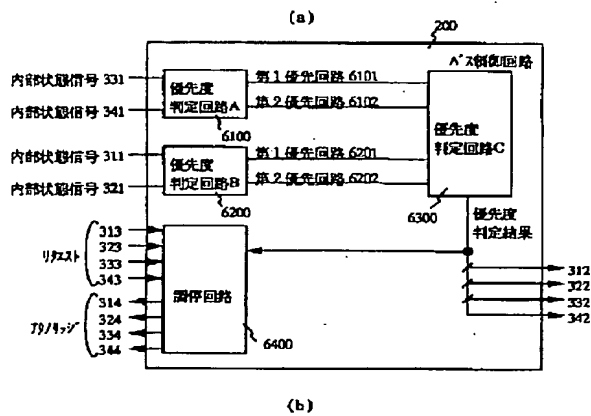
【図5】

5



【図6】

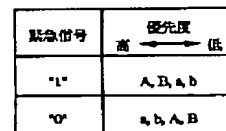
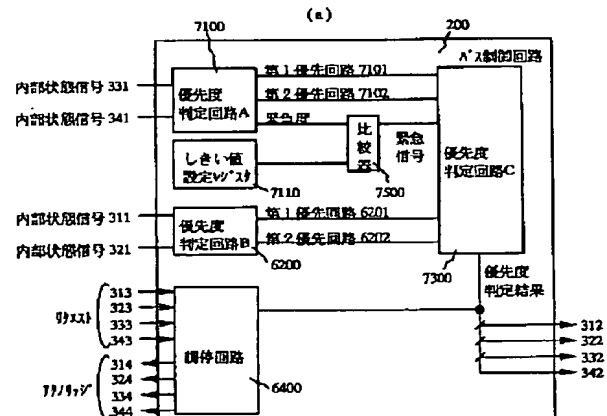
圖 6



- A: 優先度判定回路Aの第1優先回路
B: 優先度判定回路Aの第2優先回路
a: 優先度判定回路Bの第1優先回路
b: 優先度判定回路Bの第2優先回路

【図7】

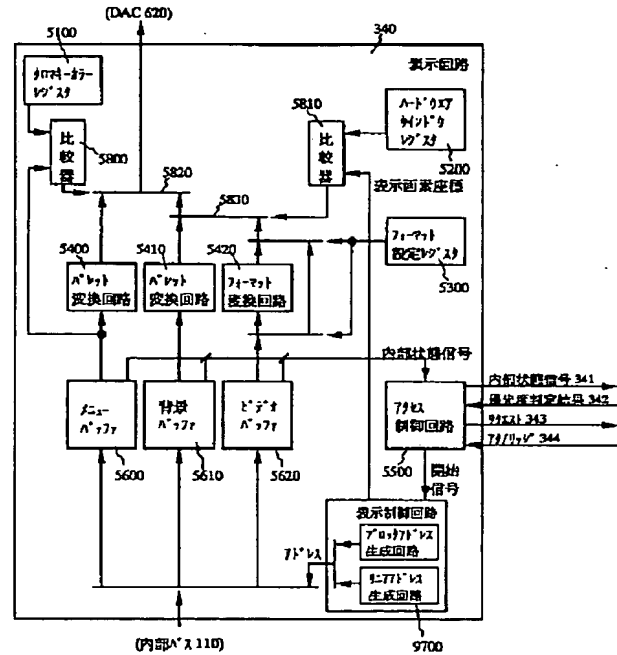
7



- A: 優先度判定回路Aの第1優先回路
B: 優先度判定回路Aの第2優先回路
a: 優先度判定回路Bの第1優先回路
b: 優先度判定回路Bの第2優先回路

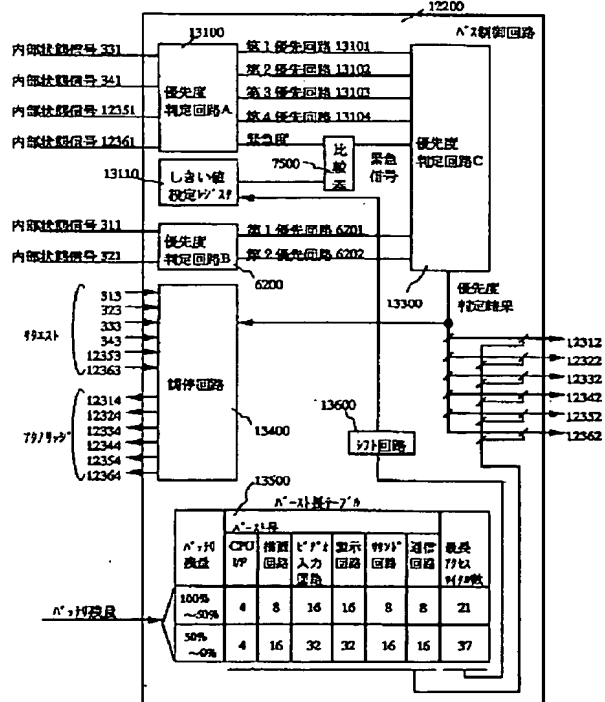
【図9】

9



【図 13】

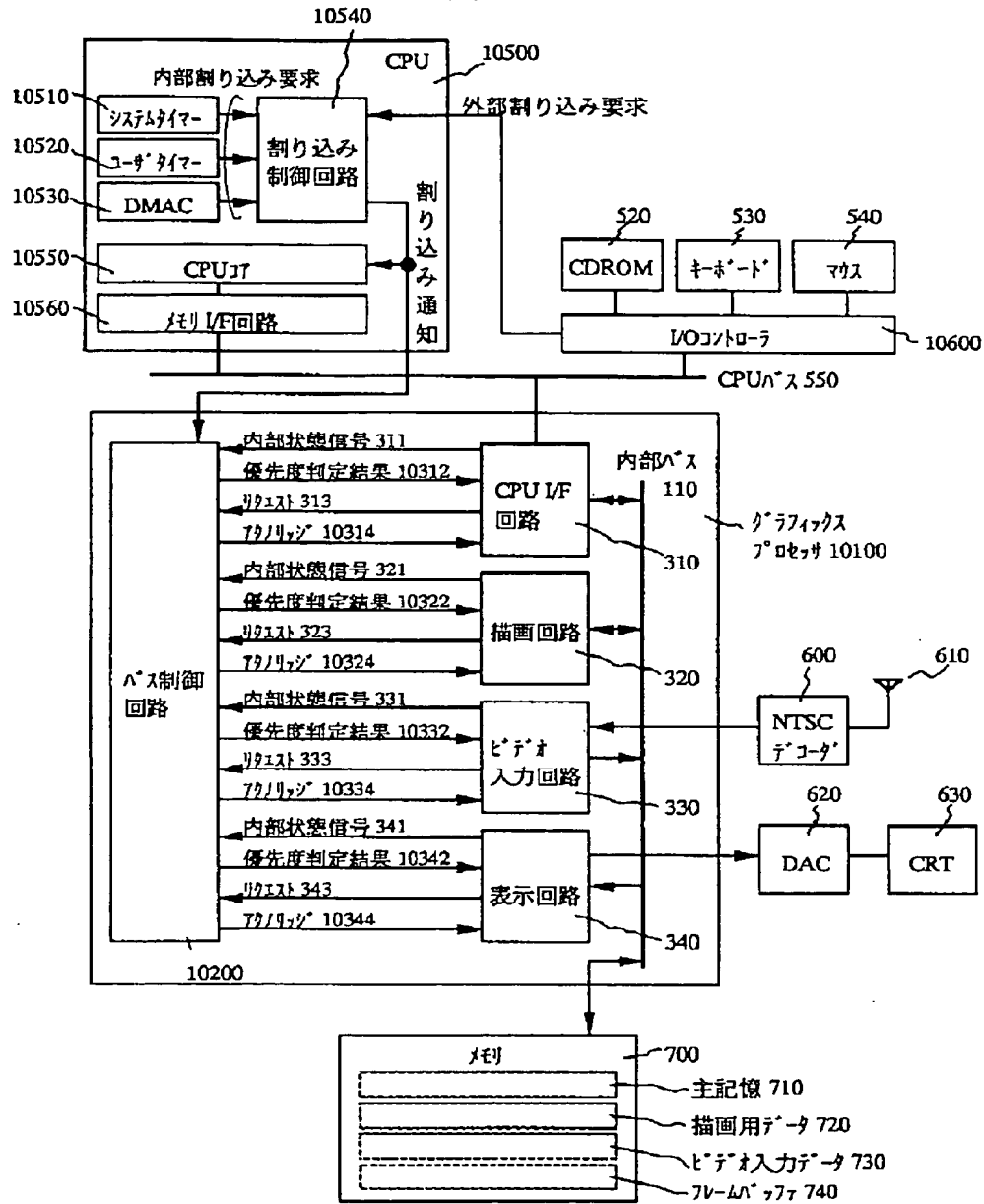
13



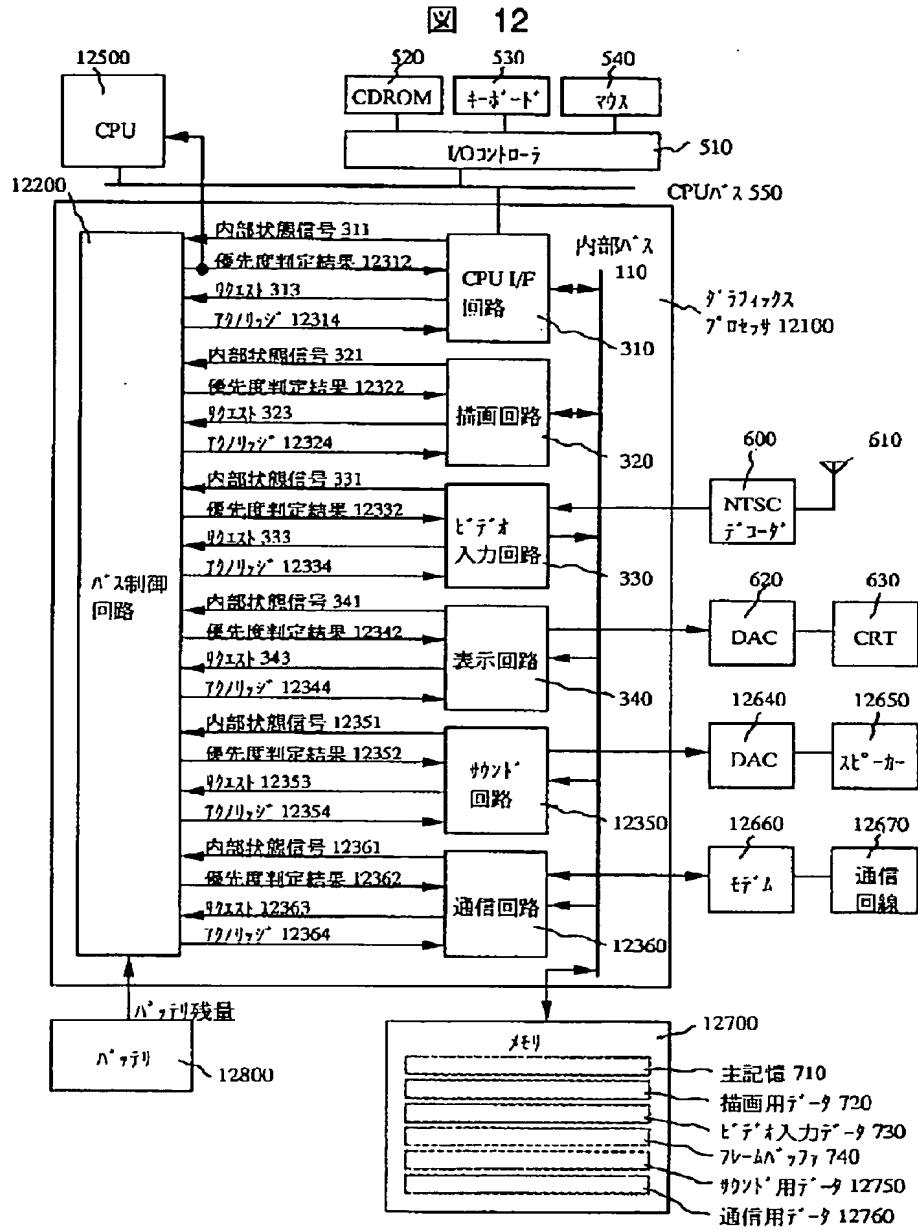
A: 優先度判定回路Aの第1優先回路 a: 優先度判定回路Bの第1優先回路
B: 優先度判定回路Aの第2優先回路 b: 優先度判定回路Bの第2優先回路

【図10】

図 10

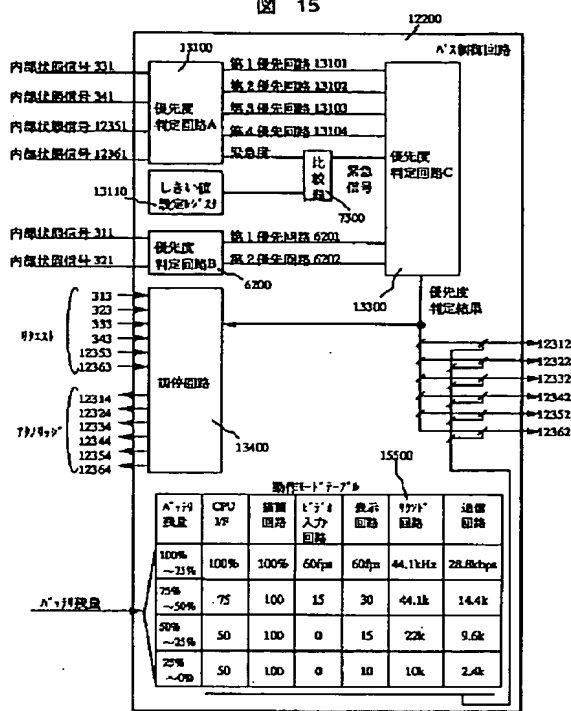


【図12】



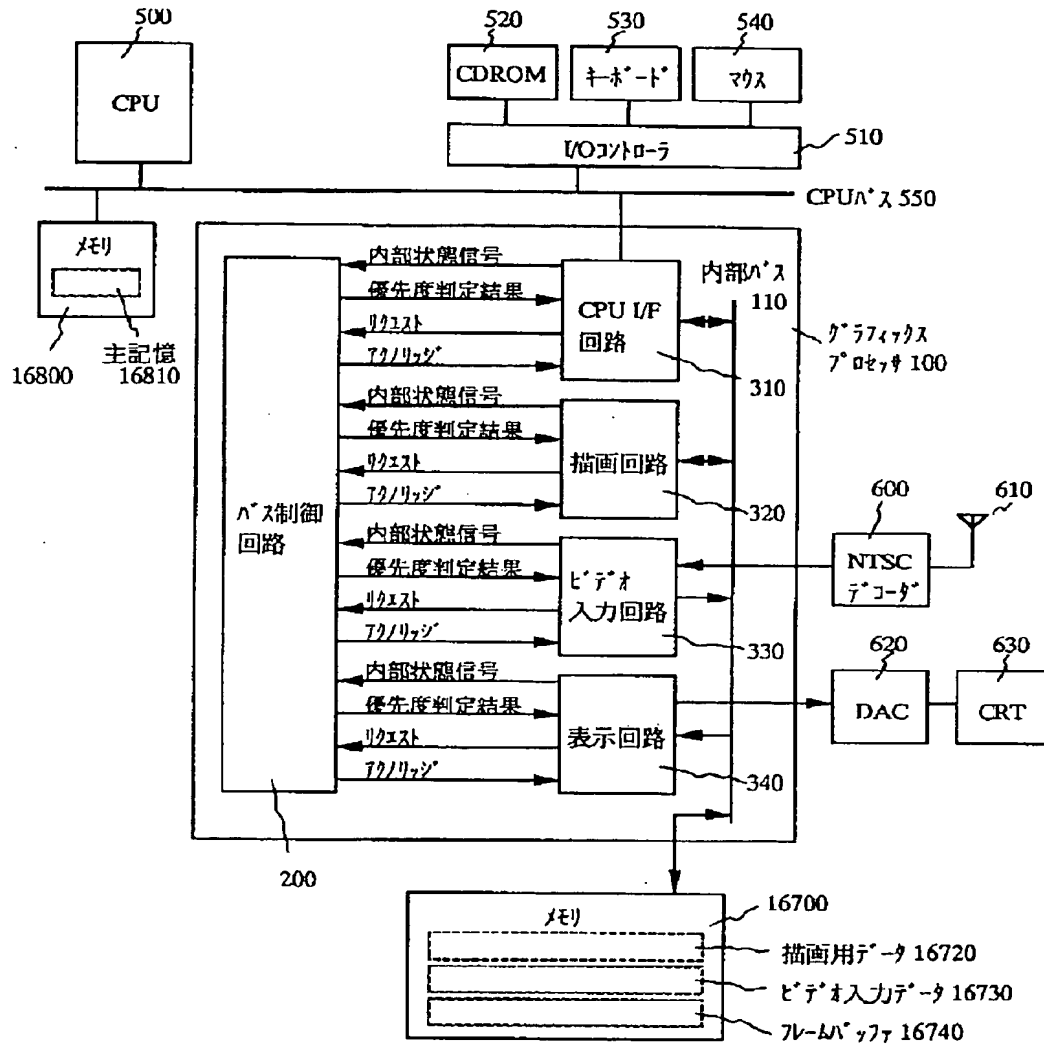
【図 15】

15



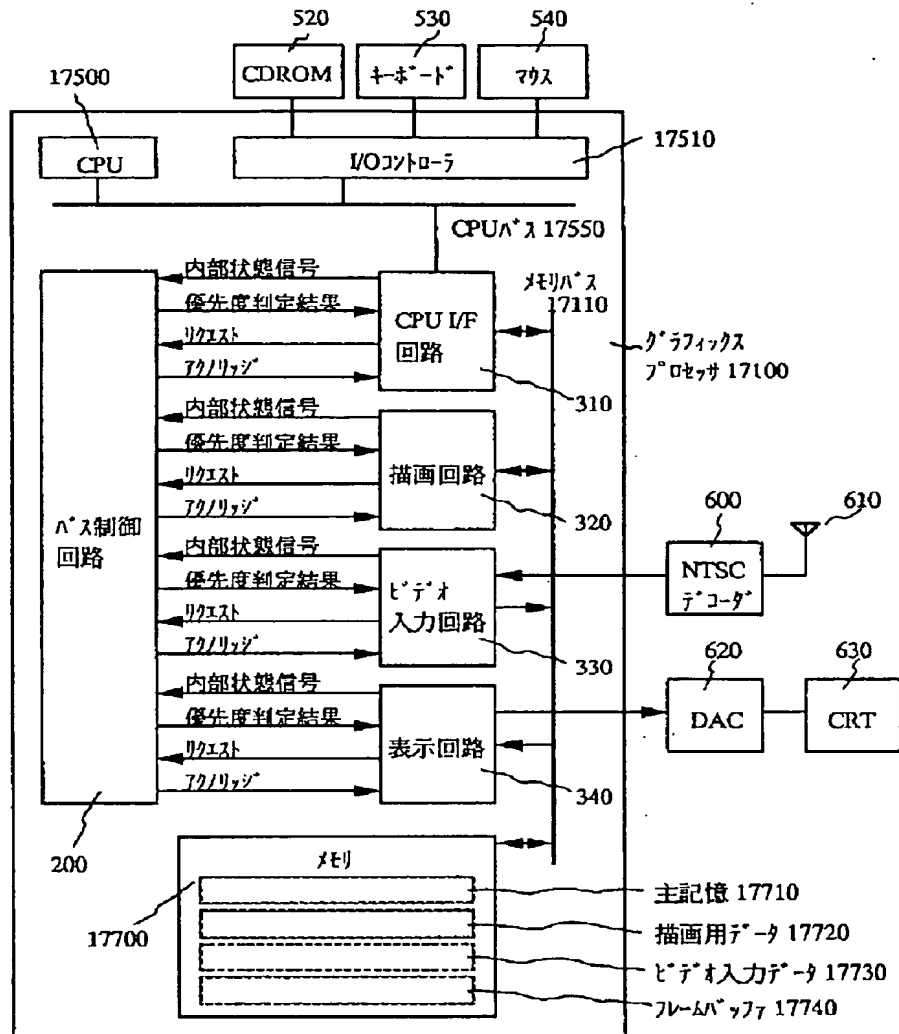
【図16】

図 16



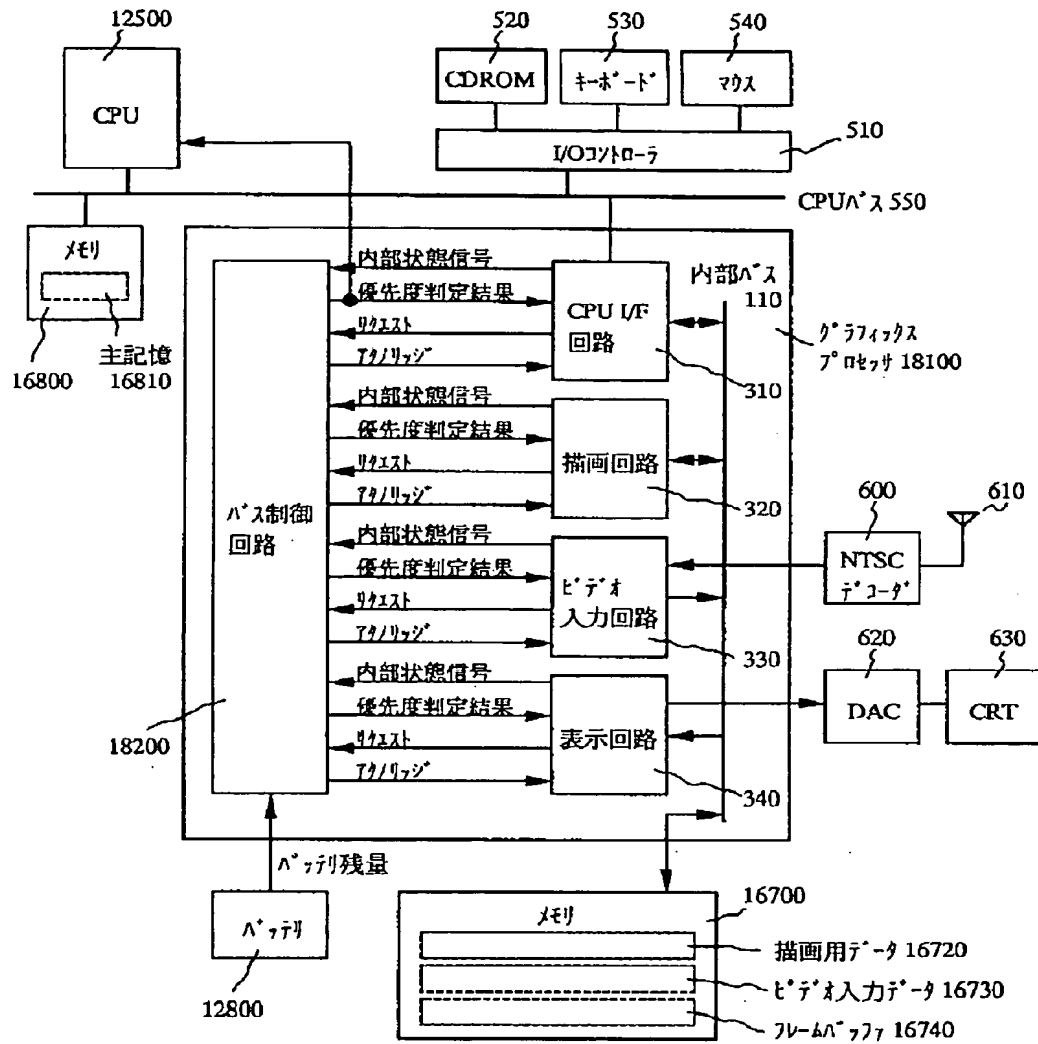
【図17】

図 17



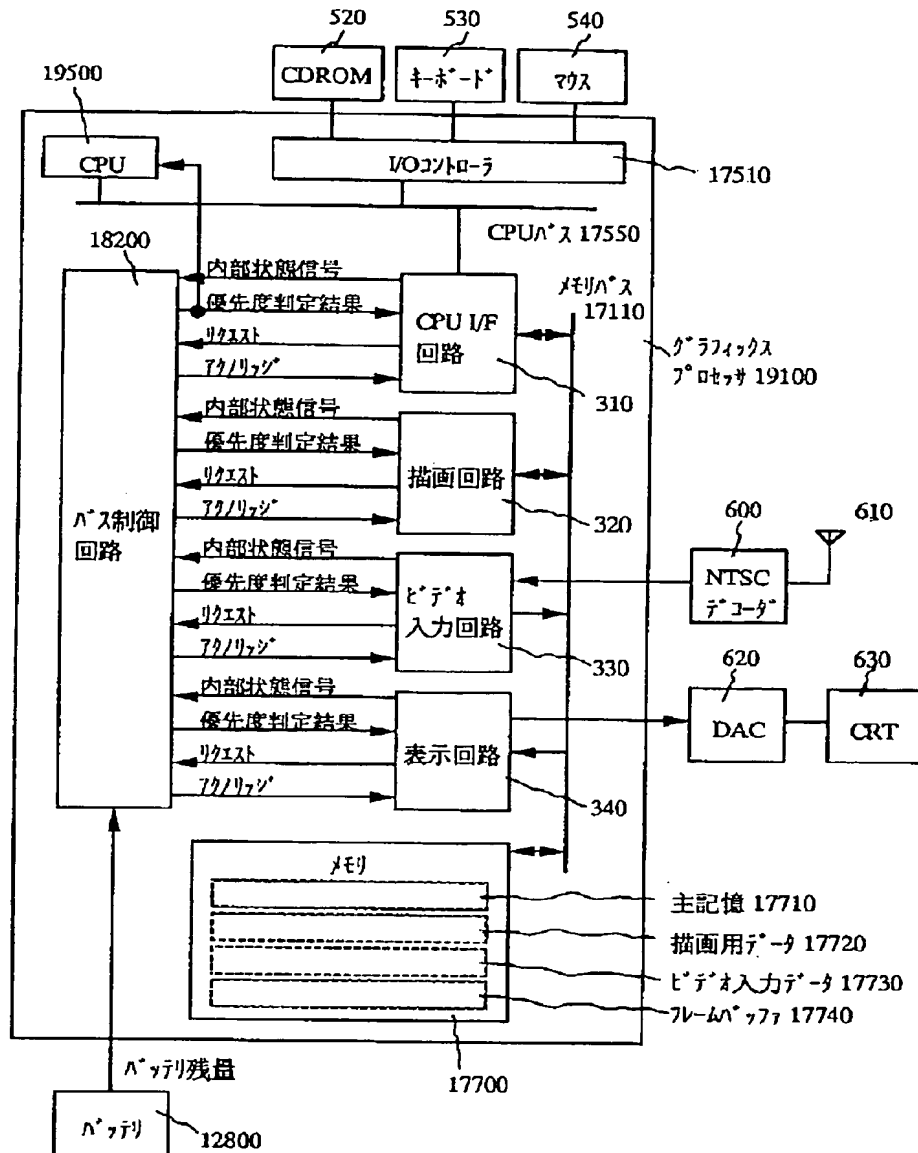
【図18】

図 18



【図19】

図 19



フロントページの続き

(72)発明者 桂 晃洋
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 中塚 康弘
東京都小平市上水本町五丁目20番1号 株
式会社日立製作所半導体事業部内

(72)発明者 山岸 一繁
東京都小平市上水本町五丁目20番1号 株
式会社日立製作所半導体事業部内

W-CDMA動画像端末用 MPEG-4マルチコーデックLSI

MPEG-4 Multi-Codec LSI for W-CDMA Image Terminals

伊 藤 博 之*¹
Hiroyuki Ito

九 郎 丸 俊 一*²
Shunichi Kurōmaru

樋 田 博 明*³
Hiroaki Toida

要 旨

従来のMPEG-4 LSIでは、複数の動画像を同時に処理する等のアプリケーションを実現できなかった。また、動画像の拡大・縮小や任意配置など多彩な表示機能を実現するには、周辺回路用LSIを別途開発して実装する必要があり、端末の高機能化、小型化、低消費電力化にとって大きな負担となっていた。

本MPEG-4マルチコーデックLSIは、MPEG-4規格に準拠した複数動画像の同時再生・表示を行うマルチコーデック機能やMPEG-4コアプロファイルを実現する。

画像圧縮・伸張において、演算量が大きな処理や定型的な処理を専用回路(ハードエンジン)で、非定型的な処理をDSPコアで実現することにより、性能向上をはかった。また、DRAMのオンチップ化、クロックゲーティング技術により、低消費電力を達成した。さらに、独自のエラー回復技術、レート制御、ノイズ除去技術により、画質の向上をはかった。

これにより、移動機によるTV電話機能やインターネットの動画コンテンツの閲覧等、多様な動画像処理を実現した。

Abstract

It has been impossible up to now for conventional MPEG-4 LSIs to provide advanced processing functions, such as the simultaneous processing of multiple moving images. To achieve a range of advanced display functions, such as scaling and arbitrary layout of moving images, extra peripheral LSIs need to be separately developed and mounted. This significantly impedes the introduction of improved features, miniaturization, and reduction of power consumption.

This new MPEG-4 multi-codec LSI can simultaneously play or display multiple moving images, and a function for processing MPEG-4 Core Profiles.

We have improved image compression/decompression performance by assigning high-volume or routine operations to 'hard engine' dedicated circuits and assigning non-routine operations to a DSP core. In addition, power consumption has been reduced by using an on-chip DRAM and clock-gating technologies. Image quality has also been improved using our own technologies for error recovery, rate control, and noise removal.

These features enable various types of image processing to be added, such as a videophone function in mobile phones and browsing of Internet content, including moving images.

*1 松下通信工業(株) モバイルコミュニケーションカンパニー
Mobile Communications Company, Matsushita
Communication Industrial Co., Ltd.

*2 半導体社 ITシステムLSI開発センター
Information Technology System LSI Development Center,
Semiconductor Company

*3 マルチメディア開発センター
Multimedia Development Center

1 緒 言

W-CDMA、PDC、PHS等の移動機向けに、世界で初めて複数動画像の同時圧縮・伸張(マルチコーデック)やMPEG-4コアプロファイルに対応したMPEG-4マルチコーデックLSIを開発した。本稿では、本開発LSIの特徴と活用技術について解説する。

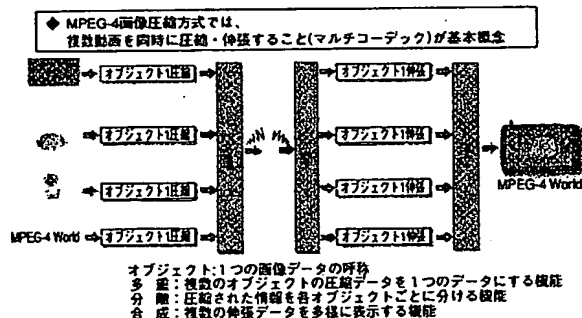
2. 背景 (要素技術MPEG-4について)

MPEG-4は、AVとコンピュータを通信に融合させる符号化規格として、1999年4月に国際標準となった。以下、MPEG-4の特徴を従来の圧縮規格として良く知られているMPEG-2 (1995年11月国際標準化) と対比させながら解説する。まず、MPEG-2は、主として2~30Mbpsの伝送レートを扱っており、DVD等の蓄積系やBSデジタル等の放送系に実用化されている。これに対して、MPEG-4は、伝送レートが10kbps~384kbpsという低レートでの伝送を主なターゲットとして、移動通信端末やインターネット動画配信への応用が進められている。

このように、MPEG-4は、MPEG-2とは異なった分野への応用を目指しており、

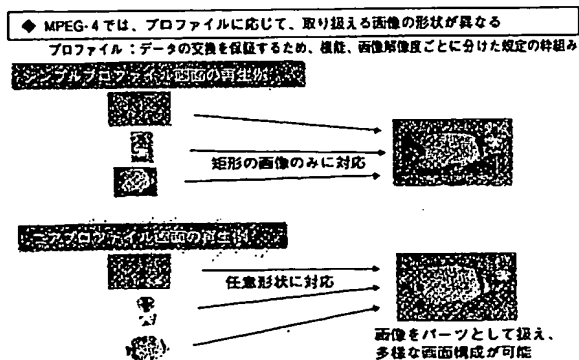
- (1) 複数動画を同時に扱うことが可能
- (2) 低レートで伝送可能
- (3) 伝送エラーへの強耐性

といったMPEG-2にない特徴を有している。ここでは、とくに複数動画の同時圧縮・伸張について述べる。MPEG-4画像圧縮では、複数動画を同時に圧縮・伸張すること、いわゆるマルチコーデックを基本概念としている。



第1図 複数動画の圧縮・伸張

Fig. 1 Concept of multi-codec. (encode and decode)



第2図 MPEG-4の各種プロファイル

Fig. 2 MPEG-4 visual profile. (simple profile, core profile)

第1図に示すように、圧縮側では、背景(海)、魚、キャラクター、文字というオブジェクトごとに圧縮することができる。この圧縮されたデータは、さらに1つのビットストリームに多重化し、圧縮側から対局(伸張側)へと送信する。一方、伸張側では、対局(圧縮側)から送信されたビットストリームを受信し、まずオブジェクトごとのデータに分離する。このオブジェクトごとのデータをそれぞれ伸張した後、1枚の画面として合成し、最終の表示画像を完成することになる。また、MPEG-4には、シンプルプロファイルとコアプロファイルというプロファイルがあり、取り扱うことのできる画像の形状が異なっている。第2図に示すように、シンプルプロファイルは、従来方式と同様、矩形のオブジェクトを扱うのに対し、コアプロファイルでは、任意形状のオブジェクトを扱うことができる。図に示すように、任意形状に対応したコアプロファイルでは、各オブジェクトをパーツと位置づけることができる。その各オブジェクトをユーザが選択、合成することにより、ユーザの好みに応じた画面を表示させることが可能となる。これは、MPEG-2等の従来方式にはない機能であり、AVとコンピュータの融合という新たなサービスを予想させる。

3. MPEG-4マルチコーデックLSI

3.1 特徴

本開発のMPEG-4マルチコーデックLSIは、以下に示す特徴を有している。

- (1) 世界で初めて、最大で2系統のMPEG-4圧縮処理、4系統のMPEG-4伸張処理の同時動作を1チップで実現
- (2) 世界で初めてMPEG-4コアプロファイルの伸張機能に対応
- (3) DRAM および各種ビデオインタフェース回路を内蔵し、1チップで移動機の動画像処理を実現
- (4) マルチコーデック時におけるMPEG-4シンプルプロファイルの圧縮・伸張処理で、1系統当たり50mWの消費電力を実現
- (5) 無線環境下での画質劣化を低減し、かつ高画質化を実現

つぎに、上記特徴について述べる。

- 1) 世界初のマルチコーデック、MPEG-4コアプロファイルに対応

今回開発したMPEG-4マルチコーデックLSIは、MPEG-4規格の基本概念であるマルチコーデックに対応すべく、最大で2系統のMPEG-4圧縮処理、4系統のMPEG-4伸張処理の同時動作を実現した。これにより、複数の動画像の同時再生・表示等の多彩な組み合わせによる映像表現が可能となる。加えて、最大高解像度の動画像(CIF30フレーム/秒)の伸張も可能であり、多様なコンテンツ再生にも対応できる。

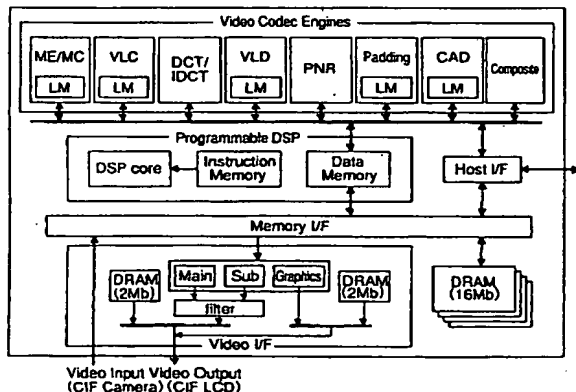
さらに、オブジェクトの任意形状の処理を可能とする MPEG-4 コアプロファイルの伸張機能に対応している。前述したとおり、これにより、オブジェクト単位のインタラクティブ表示や、また任意形状の自然画と2D、3DのCGと組み合わせて表示する等、多様なコンテンツ再生・表示をサポートできる。

これらの性能をまとめると、次のようになる。

- ・圧縮：シンプルプロファイル QCIF^{※1} 15フレーム/秒
2オブジェクト
 - ・伸張：シンプルプロファイル QCIF 15フレーム/秒
4オブジェクト
- CIF^{※1} 30フレーム/秒
- 1オブジェクト
コアプロファイル QCIF 15フレーム/秒
4オブジェクト

2) 1チップで動画処理を実現

第3図に示すように、本MPEG-4マルチコーデックLSIは、ユーザ作成のプログラムに従って動作するDSPコア部と専用ロジックからなるハードエンジン部を内蔵している。画像圧縮・伸張において、演算量が大きな処理や定型的な処理をハードエンジン部で、非定型的な処理をDSPコア部で実現することにより、性能向上をはかっている。また、本LSIは20Mbitという大容量のDRAMを内蔵している。画像処理に必要な様々な領域をすべてLSI内部に格納でき、外部にメモリを付加することなく、1チップでMPEG-4圧縮・伸張およびグラフィックスとの合成が可能である。さらに、本LSIは、CPUインタフェース、イメージセンサインタフェース、液晶ディスプレイインタフェースを内蔵しており、イメージセンサ、液晶ディスプレイとのデジタル接続により、移動通信端末の



第3図 MPEG-4マルチコーデックLSIの構成

Fig. 3 Block diagram of MPEG-4 multi-codec LSI.

※1 QCIF, CIF: 画像のサイズを示しており, CIFは352画素×288画素, QCIFは176画素×144画素から構成されている。

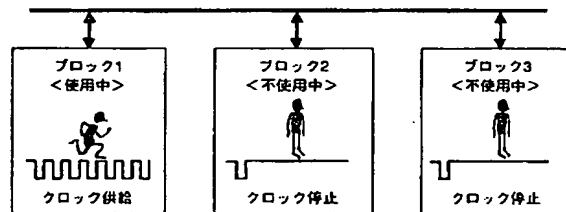
動画処理が実現できる。

3) 1系統の圧縮・伸張当たり50mWを実現

本LSIでは、各ブロックごとに供給されているクロックを個別に動作・停止できるように、クロックゲーティング技術を適用しており、第4図に示すように、不使用ブロックのクロックを停止することによって電力の浪費を防止した。また、DRAMをオンチップ化することにより、従来のチップ間のI/Oにおける消費電力を削減した。これらの技術の適用により、マルチコーデック時におけるMPEG-4シンプルプロファイルの圧縮・伸張の同時処理で、1系統当たり50mWという低消費電力化を実現することができた。

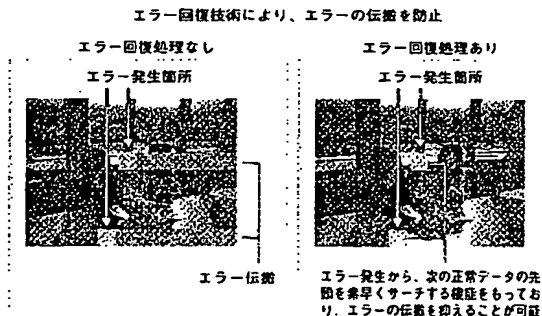
4) 無線伝送下での画質劣化を低減し、高画質を実現

本LSIでは、無線伝送時に発生したビットエラーを検出し、次の正常データの先頭を素早くサーチする機構を備えており、伝送エラーの伝搬を抑えることができ、第5図に示すように、電波受信状態の悪い環境下においても比較的良好な画質を提供することが可能なエラー回復技術を有している。また、搭載している独自のレート制御技術は、リアルタイム動画通信において重要な遅延低減を実現しつつ、動き(1秒当たりのコマ数)と画質のバランスを最適に制御することにより、高画質化をはかっている。さらに、圧縮処理により、画像処理の最小単位(8画素×8画素)の境界や物体の輪郭部分に発生する



第4図 クロックゲーティング技術

Fig. 4 Clock-gating technologies.



第5図 エラー回復技術

Fig. 5 Error image recovery technologies.

第1表 チップの主要諸元

Table 1 Major specifications of LSI.

品 番	MN1959041
電 源 電 圧	外部2.9V~3.3V、内部1.8V
消 費 電 力	50mW (マルチコーデック時におけるシンプルプロファイルの圧縮・伸張処理で1系統当たりの電力)
内部動作周波数	54MHz
プ ロ セ ス	0.18 μ m CMOS DRAM混載プロセス
パ ッ ケ ー ジ	CSP 239pin (11mm \times 11mm)
画像符号化方式	H.263 ^{※2} /MPEG-4
画像最大サイズ	CIF (352画素 \times 288画素)
対 応 モード	圧縮・伸張：シンプルプロファイル QCIF 15フレーム/秒 1オブジェクト 伸張のみ：シンプルプロファイル CIF 30フレーム/秒 1オブジェクト または QCIF 30フレーム/秒 4オブジェクト コアプロファイル QCIF 15フレーム/秒 4オブジェクト
マルチコーデック 時の最大性能	圧縮：(QCIF 15フレーム/秒 1オブジェクト) \times 2 伸張：(QCIF 15フレーム/秒 1オブジェクト) \times 4
ポスト画像処理	ノイズ除去フィルタ
その他の機能	ミラー自画像 ^{※3} 、ピクチャインピクチャ ^{※4} 、グラフィックスオーバーレイ ^{※5}

特有のノイズを独自のフィルタで除去し、更なる高画質化をはかっている。

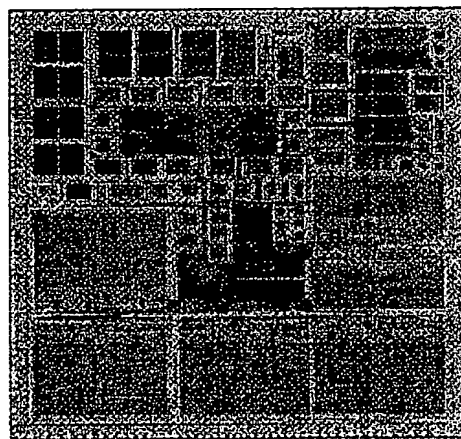
3.2 チップ諸元

今回開発したLSIは、以上のような特徴を備えており、第1表にチップの主要諸元をまとめる。また、チップ写真を第6図に示す。

3.3 MPEG-4マルチコーデックLSIの応用技術

ここでは、マルチコーデックとMPEG-4コアプロファイルの応用例を紹介する。まず、第7図にマルチコーデックの応用例を示す。本LSIを用いると、携帯テレビ電話等で、画面に双方の顔を表示しつつ、本体に蓄積した別の映像を同時に再生・表示しながら通話することも可能になる。

また、本LSIはMPEG-4コアプロファイルに対応しており、第8図に示すように、背景動画の上に、ユーザが好



第6図 チップ写真

Fig. 6 Chip micrograph.

む任意形状の動画やキャラクタ等の任意位置への配置・表示なども容易に実現できる。

4. 結 言

以上述べてきたように、今回開発したMPEG-4マルチコーデックLSIは、

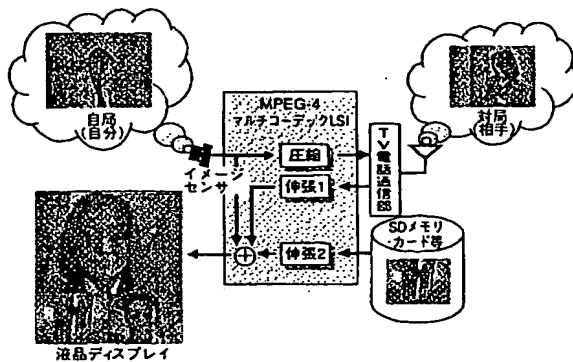
- ・マルチコーデックの実現
- ・MPEG-4コアプロファイルへの対応
- ・DRAM、各種IFの搭載
- ・低消費電力動作
- ・無線伝送下での高画質の実現

※2 H.263：ITU-T が1996年に制定した画像圧縮の国際標準規格であり、既存の電話網などの低速度の通信網からLANなどの高速度の通信網までサポートしている。

※3 ミラー自画像：イメージセンサで撮影している自画像を液晶ディスプレイに表示する際に、鏡に映しているように画面の左右を反転させる機能。

※4 ピクチャインピクチャ：対局とTV電話をする際に、対局から送られてくる画像の上に、イメージセンサで撮影している自画像を重ねて表示する機能。

※5 グラフィックスオーバーレイ：動画像の上にグラフィックスを重ねて表示する機能。



第7図 マルチコーデック応用例

Fig. 7 Example of multi-codec system.

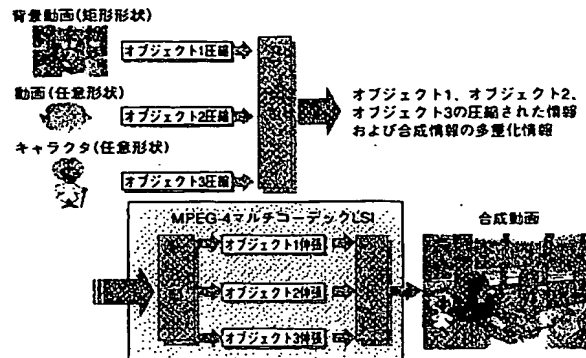
という特徴を備え、高性能でかつ低消費電力を実現した。

また、本稿では紹介しなかったが、H.324画像・音声コーデック処理の実現に向けた取組みも行っている。画像専用のLSIとしてだけではなく、画像・音声コーデックLSIとして、展開の幅を広げるのがこの取組みの狙いである。

今後は、本LSIの製品化を進め、モバイルマルチメディア市場に展開していく予定である。

参考文献

- 1) Nishikawa T., et al. : A 60MHz 240mW mpeg-4 video-phone LSI



第8図 MPEG-4コアプロファイル応用例

Fig. 8 Example of MPEG-4 core profile.

with 16Mb embedded DRAM. ISSCC Digest of Technical Papers, pp.230-231, Feb. (2000).

- 2) Kamemaru T., et al. : Media processor core architecture for realtime, bi-directional MPEG4/H.26X codec with 30fr/s for CIF-Video. Proc. of CICC'00, pp.473-476, May (2000).
- 3) Kurōmaru S., et al. : A MPEG4 programmable codec DSP with an embedded pre/post-processing engine. Proc. of CICC'99, pp.69-72, May (1999).
- 4) Hashimoto T., et al. : A 90mW MPEG4 video codec LSI with the capability for core profile. ISSCC Digest of Technical Papers, Feb. (2001).